KS. DOÃN HOA

MỘT SỐ CHƯƠNG TRÌNH

PASCAL ĐƠN GIẢN

DÙNG CHO THIẾT KẾ ĐƯỜNG Ô TÔ



NHÀ XUẤT BẢN XÂY DỰNG

KS. DOÃN HOA

MỘT SỐ CHƯƠNG TRÌNH PASCAL ĐƠN GIẢN

DÙNG CHO THIẾT KẾ ĐƯỜNG Ô TÔ

NHÀ XUẤT BẢN XÂY DỰNG
HÀ NỘI - 2004

LỜI NÓI ĐẦU

Kiến thức về ứng dụng máy vi tính trong hoạt động văn hóa - kinh tế - kỹ thuật cho mọi ngành nghề, đến ngày nay, có thể ví như biển cả mênh mông. Biển của thế giới không thay đổi, không "mở rộng" thêm, nhưng biển của ứng dụng máy tính lại mở rộng "hàng ngày" !

Nếu ví ứng dụng máy vi tính cho mọi ngành nghề như biển cả mênh mông, thì ta có thể coi phạm vi ứng dụng cho một chuyên ngành như vùng biển của một quốc gia.

Hiện có rất nhiều phần mềm, sách hướng dẫn sử dụng máy vi tính theo Windows, Excel, AutoCAD, Softdesk, SAP... đó là những hướng dẫn "cho mọi ngành nghề". Thực tế một kỹ sư chuyên ngành, chỉ có thể đọc, hiểu và vận dụng một số rất nhỏ các hướng dẫn đó (trong một tập sách dày hàng trăm trang). Sách hướng dẫn vận dụng cụ thể cho một chuyên ngành, như thiết kế - thi công đường , thì "hình như" còn ít có.

Một kỹ sư cầu đường sử dụng máy vi tính thường gặp hai loại việc:

- Lên bản vẽ thiết kế bình đồ, cắt dọc, cắt ngang đường: có thể dùng phần mềm Softdesk hoặc NOVA (của Công ty Hài Hòa).

- Tính kết cấu cầu, cống bằng phần mềm SAP. Tính ổn định mái dốc bằng Bishop. Ngoài ra, còn có một số phần mềm khác do một số cơ quan "tự biên" về tính toán độ lún, cải tạo nền đất yếu v.v..., nhưng chưa hẳn được thừa nhận và phổ cập rộng rãi.

Thực tế cũng cho thấy là: một kỹ sư "quen" sử dụng phần mềm tính toán sẽ có thể "quên" nguyên lý tính toán, công thức tính toán. Mà các phần mềm trên không phải lúc nào cũng có sẵn, có đủ để cài đặt vào mọi máy tính (ở một công ty thiết kế hay trong một công trường thi công). Ấy là chưa kể học cách sử dụng cũng không phải là đơn giản. Chính vì vậy đã có nhận xét: máy vi tính giúp kỹ sư "làm việc nhanh" nhưng cũng "làm lười, làm dốt" việc học tập tri thức mới của kỹ sư !

Cách khắc phục có thể là: các kỹ sư nên nắm vững các nguyên lý tính toán, để ở mọi nơi, mọi lúc, có thể dựa vào Excel, Pascal, Visual Basic... "tự biên tự diễn" các chương trình con phục vụ cho công việc, và theo các nguyên lý tính toán mà mình cho là hợp lý.

Tác giả viết cuốn sách này nhằm "gợi ý" cho phương pháp làm việc trên.

Một số bài toán lập trình giới thiệu chỉ bó hẹp trong phạm vi thiết kế ổn định nền đường, thiết kế thoát nước.

Sự cải tiến về máy vi tính, về chương trình phần mềm (Windows, Excel, AutoCad, SAP...) thay đổi hàng tháng, hàng năm. Do đó có e ngại: viết về máy vi tính vừa xong thì đã lạc hậu rồi ! Tuy vậy, cũng có thực tế là: các phiên bản (version) sau của máy vi tính thường theo nguyên tắc kế thừa phiên bản trước. Do đó ta có thể yên tâm: những điều viết ra theo phiên bản trước vẫn giúp ta ứng dụng theo phiên bản sau, đúng như nguyên tắc "ôn cố tri tân".

Cuốn sách này được biên soạn theo trình tự: giới thiệu lý thuyết chung, ví dụ tính toán bằng tay, sau đó lập chương trình tính toán Pascal để minh họa, đối chiếu.

Ngoài ra, trong phần Phụ lục có giới thiệu sơ lược cách sử dụng phần mềm NOVA và SAP 2000.

Do trình độ có hạn, những chương trình viết ra chỉ ở mức đơn giản. Chắc rằng các bạn đọc, đặc biệt là các kỹ sư trẻ, còn có nhiều cách lập trình khác hay hơn nhiều.

Mong độc giả góp ý, lượng thứ cho các thiếu sót khó tránh khỏi khi viết tập sách này.

Tác giả

Chương 1

CƠ CẤU - ỨNG DỤNG CHUNG MÁY VI TÍNH

1-1. LỊCH SỬ PHÁT TRIỂN MÁY TÍNH

1) Khái quát chung

Ở Việt Nam, từ những năm 1960 - 1970 còn dùng "máy tính" là thước Logarit. Năm 1970 - 1980 là máy tính quay tay (như máy NISA của Tiệp); một số cơ quan thiết kế còn dùng cả "bàn gẩy" của Trung Quốc. Mãi đến những năm 90, máy vi tính cá nhân (PC- Personal Computer) mới từng bước thâm nhập ứng dụng vào Việt Nam, lúc đầu là máy 286 màn hình đen trắng, sau đến 386 màn hình màu. Tháng 11 năm 1997 bắt đầu kết nối Internet.

Trên thế giới, chặng đường phát triển của máy vi tính, từ bước đi dò dẫm ban đầu khá sớm, nhưng cũng rất gian khổ. Từ 1820, nhà toán học người Anh Charles Babgage đã thiết kế chiếc máy tính đầu tiên, với ý tưởng thực hiện các phép tính số học, nhưng chưa hoàn thành. Đến 1936, một nhà toán học Anh khác, Alan Turing mới chế tạo thành công chiếc máy tính đầu tiên, có tên gọi "Colossus", được dùng để phá mật mã của Đức và giữ bí mật cho đến năm 1970.

Sự đóng góp của Mỹ vào lĩnh vực máy tính điện tử đầu tiên là máy Mark I do Đại học Tổng hợp Harvard và hãng IBM thiết kế từ năm 1930, dùng rơ le chuyển mạch điện thay thế cho cơ cấu chuyển mạch cơ khí. Máy này được dùng để tính đường đạn cho vũ khí của Hải quân Mỹ.

Đến 1940, bắt đầu hình thành một "cơn sốt" chế tạo máy tính giữa các nhà khoa học ở cả 2 bên bờ Đại Tây Dương. Năm 1945, các kỹ sư Mỹ Mauchly và J. Presper Eckert cùng trường Đại học Pennsylvania (Philadelphia) đã chế tạo thành công chiếc máy tính ENIAC (Electrolic Numerical Integrator And Computer), một "con quái vật" gồm 18.000 chiếc bóng chân không và nặng tới 27 tấn!

Có lẽ vì vậy, một trong những con người "thông minh" như ông Thomas Watson, Chủ tịch hãng IBM năm 1943 đã phát biểu: *"Tôi nghĩ rằng thị trường thế giới chỉ cần khoảng 5 chiếc máy vi tính"*! Câu nói đó đã đi vào lịch sử như một trong những đánh giá sai lầm lớn nhất của mọi thời đại.

Tuy vậy, nhiều nhà khoa học, công ty kinh doanh vẫn sớm nhận ra tác dụng to lớn, tương lai phát triển của máy tính và đưa máy tính vào công việc kinh doanh. Hãng IBM đã nhanh chóng nắm được vai trò độc tôn sau khi chế tạo thành công những chiếc máy tính nhỏ hơn và có tốc độ nhanh hơn. Năm 1981, hãng này bước vào địa hạt máy tính cá nhân (PC), tạo ra một thị trường mà IBM được coi là tiêu chuẩn không chính thức.

Máy tính cá nhân chỉ thực sự phổ cập khi Bill Gates, từ 1975 thành lập công ty Microsoft và bắt đầu bán những phần mềm lập trình cho loại computer rẻ tiền Altair.

Tuy vậy, sự phát triển nhanh chóng máy tính cá nhân cũng là "ngoài sức tưởng tượng" của không ít người, trong đó có ông Ken Olsen- Chủ tịch kiêm người sáng lập ra hãng Digital Equipment Corp, năm 1977 đã phát biểu: *"Chẳng có lý do nào khiến người ta muốn có một chiếc computer trong nhà"*.

Trong khi đó, ngay từ cuối thập kỷ 80, bắt nguồn từ mạng máy tính nội bộ của quân đội Mỹ, mạng Internet đã ra đời và phát triển nhanh chóng trên toàn thế giới. Chỉ trong vòng 10 năm, Internet tạo nên một thế giới huy hoàng không bị chia cắt, làm đảo lộn cuộc sống của nhân loại, cuốn hàng tỷ người sinh hoạt và làm việc theo thói quen mới.

Nhưng dàu sao cũng còn một "thế giới khác" của các nước chưa phát triển thuộc khu vực rộng lớn trên trái đất như Châu Phi, Nam Á... Cuộc cách mạng thông tin của Internet chưa "thấm tới" mảnh đất này. Các nước đó đã nghèo sẽ tiếp tục nghèo, còn các nước giàu tiếp tục giàu hơn. Cuộc chiến thu hẹp khoảng cách giàu nghèo nằm trong cuộc chiến nắm bắt thông tin. Ai nắm được thông tin, người đó chiến thắng.

2) Cấu tạo máy vi tính cá nhân (PC – Personal Computer)

Thuật ngữ *"máy tính"* quen dùng hiện nay thực chất là chỉ loại *máy tính số* (Digital Computer). Máy tính cá nhân PC, quen gọi là *máy vi tính*, là máy dùng cho một người.

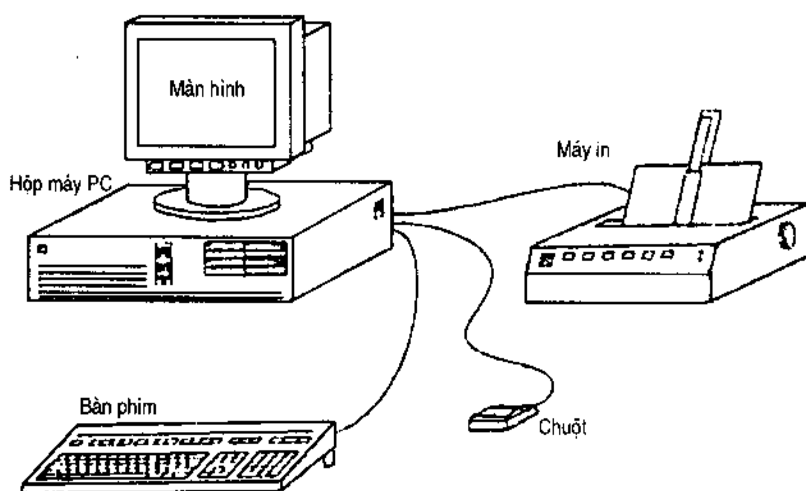
a. Cấu tạo chung của máy vi tính

Hình 1-1 giới thiệu khái quát các thành phần cấu tạo chính một máy vi tính: hộp máy PC chứa bộ vi xử lý (nên còn gọi là hộp CPU), ổ đĩa cứng, đĩa mềm, các mạch ghép nối...Bàn phím *(keyboard)*. Màn hình *(monitor)*. Chuột *(mouse)*. Máy in *(printer)*.

Hiện nay, còn dùng nhiều thiết bị ngoại vi khác nữa như: loa nghe nhạc, joystick (bàn phím trò chơi), modem nối Internet, webcam (ghi hình qua mạng)...

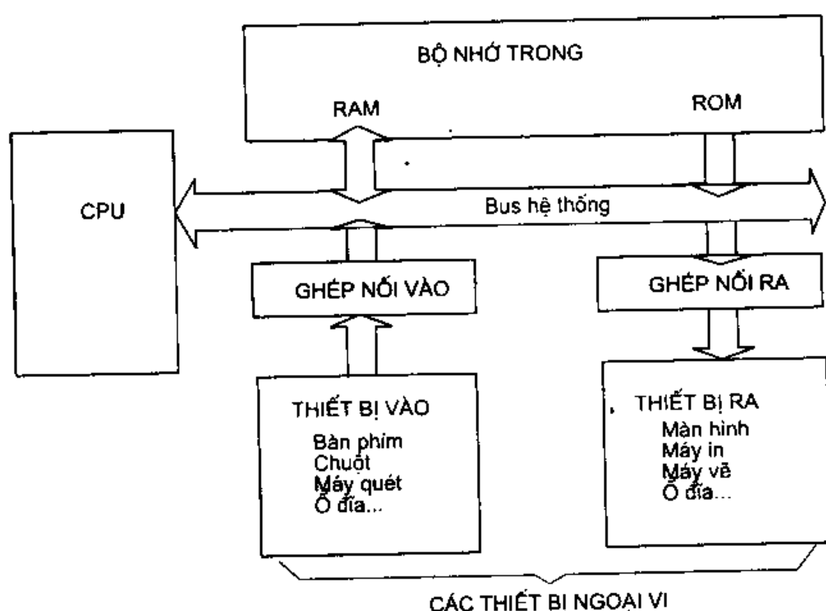
Các thành phần hữu hình trên được gọi chung là *phần cứng*.

Để người sử dụng làm việc được với PC, cần phải có các *phần mềm*, là các chương trình lập sẵn để người sử dụng dùng để điều khiển máy tính hoạt động theo yêu cầu mong muốn. Phần mềm quan trọng nhất là hệ điều hành (như DOS, Windows, Macintosh, Linux...). Sau đó đến các phần mềm ứng dụng khác như Office (gồm Word, Excel, thường gọi là phần mềm văn phòng), Norton Commander (NC), Pascal (TP- lập trình), Autocad (vẽ kỹ thuật), Softdesk (thiết kế kiến trúc, cầu đường), SAP (tính kết cấu), Lạc Việt từ điển, Antivirus...



Hình 1-1: *Cấu tạo chính một máy vi tính.*

Hình 1-2 giới thiệu sơ đồ khối chức năng một hệ thống máy vi tính, gồm các khối như sau:



Hình 1-2: *Sơ đồ khối chức năng của một hệ thống máy vi tính.*

- *Khối xử lý trung tâm CPU (central processing unit):* dùng để thu thập và cho chạy các lệnh. CPU máy PC thường cấu tạo một chíp vi mạch gọi là bộ *vi xử lý* µP (microprocessor). Có thể nói đây là quả tim của máy vi tính. Đặc trưng kỹ thuật điển hình của bộ vi xử lý là tốc độ truyền dữ liệu, đo bằng MHz (mê ga hec).

Bộ vi xử lý quen thuộc nhất là Pentium, đã được cải tiến không ngừng: năm 1997 là PentiumII có tốc độ 266 MHz, tháng 2-1999 ra đời Pentium III có tốc độ 500 MHz, đến tháng 11-2003 đã lên tới 3,5 GigaHz (Pentium IV).

- *Bộ nhớ (memory):* dùng để lưu trữ các lệnh và dữ liệu cho bộ xử lý. Gồm 2 loại: bộ nhớ trong (được tạo bởi các vi mạch nhớ bán dẫn, điển hình nhất là ổ đĩa cứng – hard disk) và bộ nhớ ngoài (được tạo bởi các môi trường nhớ khác như đĩa từ, đĩa quang).

Thông số kỹ thuật của bộ nhớ đo bằng Mbyte hoặc Gigabyte. Dung lượng nhớ này cũng không ngừng được cải tiến: năm 1997 ổ đĩa cứng một máy PC mới phổ cập 2 – 3 Gigabyte thì đến 2003 đã phổ cập 20 – 30 Gigabyte. Ổ đĩa mềm 3,5 inches quen dùng có dung lượng 1,2 Mbyte, nhưng ổ CDRom có dung lượng tới 600 Mbyte.

- *Các thiết bị ngoại vi:* gồm các thiết bị vào ra (I/O) dùng để nhập hoặc xuất các dữ liệu. Bàn phím, chuột, máy quét... thuộc loại thiết bị vào. Màn hình, máy in ... thuộc loại thiết bị ra. Các ổ đĩa ở bộ nhớ ngoài có thể coi là thiết bị vừa vào vừa ra.

Các thiết bị ngoại vi được liên hệ với CPU qua các *mạch ghép nối vào/ra* (I/O – in/out interface). Mạch này cho phép nối 2 bộ phận độc lập nhằm làm cho chúng có thể tương hợp và thông tin được vơí nhau.

- *Bus hệ thống:* là một tập hợp các đường dây mà qua đó CPU có thể liên kết với các bộ phận khác.

Bộ nhớ thường được chia thành từng ô nhớ nhỏ như từ hay *byte* (1 byte = 8bit, 1 từ = 2 byte). Mỗi ô nhớ đó cũng như một thiết bị vào/ra được gán cho một địa chỉ (address) để CPU có thể định vị khi cần đọc hay viết dữ liệu lên nó.

Máy tính phải có một mạch tạo các xung điện gọi là đồng hồ hệ thống (system clock) để duy trì hoạt động và đồng bộ hoá CPU cùng các bộ phận liên quan với nhau. Tần số đồng hồ hệ thống này quyết định tốc độ hoạt động của CPU.

Toàn bộ hệ thống máy tính hoạt động nhờ nguồn điện xoay chiều (220V) hoặc acquy. Do nhiều thiết bị ngoại vi cần nối trực tiếp vào ổ cắm điện nên hệ thống dây cắm khá phức tạp, cần bố trí khoa học, ổ cắm vững chắc để không xảy ra sự cố làm mất dữ liệu.

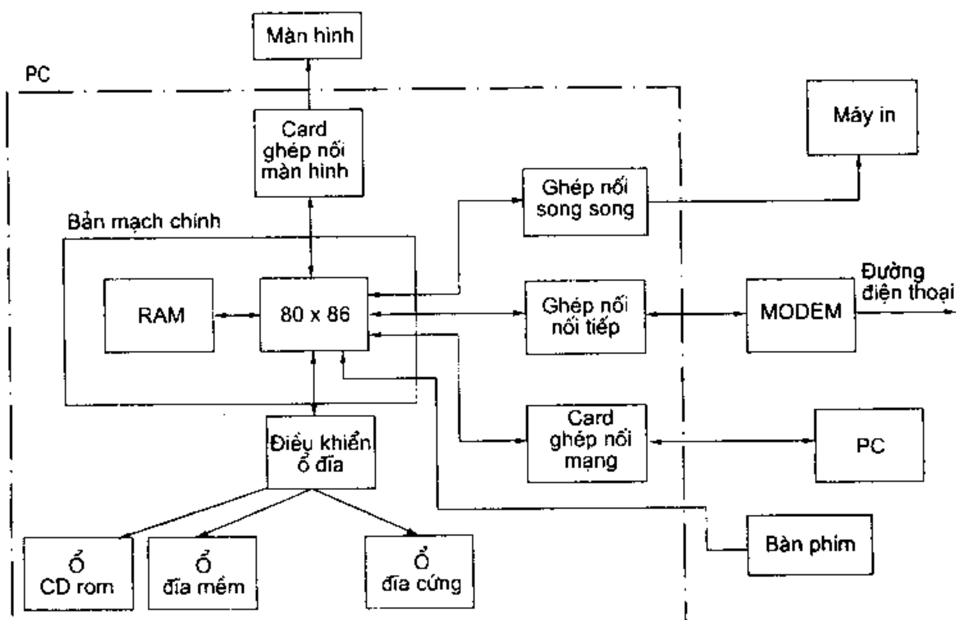
Quá trình sử dụng PC có thể theo trình tự khái quát như sau: mở máy – vào phần mềm muốn sử dụng (thí dụ Excel), lập một bảng tính toán, save - đặt tên file để lưu giữ vào ổ cứng, hoàn thành thoát ra khỏi excel, tắt máy.

Do đó, "bài học đầu tiên" khi sử dụng máy vi tính luôn là: mở máy, tắt máy, vào phần mềm ứng dụng, thoát ra khỏi phần mềm.

Máy vi tính giúp ta tăng nhanh tốc độ làm việc rất nhiều, nhưng cũng hay hỏng vặt, tạo nên nguy cơ khó chịu nhất, đáng tiếc nhất là mất dữ liệu mà ta đã dày công thiết lập, lưu trữ qua hàng tháng, hàng năm. Cho nên có người đã nhận xét: máy vi tính "đỏng đảnh như một cô gái đẹp vậy !"

b. Cấu hình bên trong máy vi tính

Hình 1-3 giới thiệu khái quát các cấu hình bên trong một hộp máy PC nối với các thiết bị ngoại vi và máy PC khác trong mạng.



Hình 1-3: Sơ đồ khối của một PC với các thiết bị ngoại vi

Tóm tắt các cấu tạo trong máy như sau:

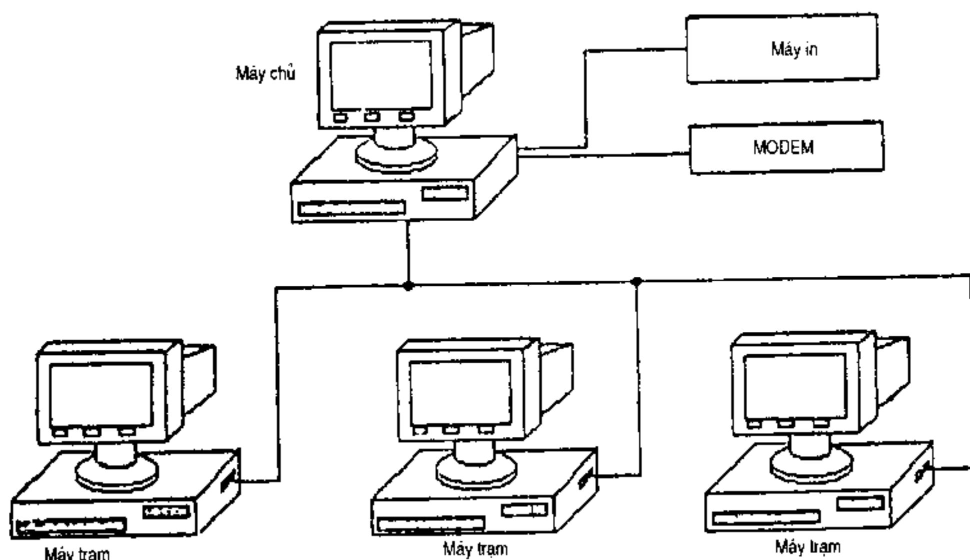
- Bộ vi xử lý 80 × 86
- Bản mạch chính (Mainboard)
- Ổ đĩa cứng (HDD - Hard Disk Drive)
- Ổ đĩa mềm (FDD - Floppy Disk Drive)
- Ổ đĩa quang (CDRom – Compact Disk Read Only Memory)
- Chíp nhớ tạm thời RAM (Random Access Memory)
- Mạch ghép nối các thiết bị vào ra Bus
- Các khe cắm mở rộng (Expansion Slots)

3) Mạng máy vi tính

Có 2 loại:

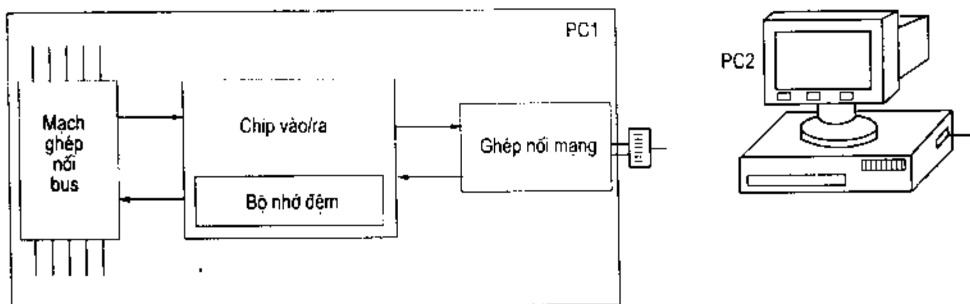
- *Mạng rộng (WAN – Wide Area Network)*: là mạng nối nhiều mạng cục bộ LAN trong phạm vi một tỉnh hay một nước. Internet là mạng liên kết toàn thế giới, giúp ta truy nhập, trao đổi số liệu với mọi người trên thế giới. Một dạng trao đổi phổ thông nhất là thư điện tử E-mail.

- *Mạng cục bộ (LAN – Local Area Network)*: dùng trong một phạm vi hẹp (như một cơ quan), xem hình 1-4. Trung tâm của LAN là một máy chính gọi là máy chủ (server). Máy này quản lý tất cả các dữ liệu chung cho toàn các nút mạng. Qua các đầu nối, máy chủ nối với các máy tính khác (được gọi là các trạm – workstation) hoặc các thiết bị ngoại vi như máy in, modem...



Hình 1-4: *Mạng cục bộ LAN*

Mỗi bản mạch ghép nối mạng có một chíp vào/ra trong đó có bộ nhớ đệm để lưu trữ tạm thời các thông tin vào/ra. Bản mạch này được cắm vào các khe cắm mở rộng của mỗi máy trên mạng, như sơ đồ hình 1-5.



Hình 1-5: *Bản mạch ghép nối mạng*

1-2. ỨNG DỤNG MÁY VI TÍNH

1) Ứng dụng chung

Ứng dụng máy vi tính trong hoạt động xã hội hiện nay rất phong phú, bao gồm mọi quốc gia, mọi ngành nghề, mọi thời gian. Khái quát như sau:

- Đơn giản phổ thông nhất là ứng dụng trong công tác văn phòng (Word, Excel...): lập các văn bản, tính toán số liệu, thống kê xử lý số liệu trong Excel ...

- Đồ hoạ: lập các bản vẽ trang trí cũng như thiết kế kỹ thuật. Trong thiết kế kiến trúc, với ứng dụng 3D cho phép tạo các bản vẽ phối cảnh rất nhanh và sinh động

- Giải trí đa phương tiện (Multimedia): nghe nhạc, xem phim, chơi games, chat, gặp mặt qua mạng với thiết bị webcam...

- Lưu trữ, trao đổi dữ liệu qua mạng Internet, E-mail...

- Thương mại điện tử: Kinh doanh, quảng cáo qua mạng

- Có thể lập cả "chính phủ điện tử"

Nhưng bao trùm lên tất cả, là công việc lập trình để có các phần mềm phục vụ cho các ứng dụng trên.

2) Phần mềm

Như đã nêu ở trên, phần mềm (Software) là thành phần kết nối người sử dụng với máy vi tính, trong đó quan trọng nhất là hệ điều hành. Sau đến các phần mềm ứng dụng khác.

Hệ điều hành là chương trình điều khiển chủ đạo đối với một máy tính, dùng để quản lý các chức năng nội trú của máy tính, và cung cấp những phương tiện để kiểm soát các hoạt động của máy đó. Những hệ điều hành máy tính cá nhân phổ biến nhất hiện nay gồm có MS-DOS, Microsoft Windows, Macintosh ...

Khái quát chung như sau:

a. MS-DOS (Microsoft Disk Operation System)

Hệ điều hành tiêu chuẩn một người sử dụng của các máy tính IBM và tương thích IBM, ra đời vào năm 1981, MS-DOS (Microsoft Disk Operating

System) đã được hãng IBM đưa ra tiếp thị với tên gọi PC-DOS; hai hệ này hầu như không thể phân biệt lẫn nhau. MS-DOS có nguồn gốc trong CP/M - hệ điều hành cho các máy tính 9 bit phổ biến trong những năm cuối 1970. Phiên bản nguyên thuỷ mà về sau trở thành MS-DOS là do một hãng nhỏ ở Seattle biên soạn với mục đích thực nghiệm. Vì Microsoft nhận hợp đồng biên soạn cho IBM một hệ điều hành để dùng với máy tính IBP-PC nên Microsoft đã mua lại và phát triển chương trình đó. MS-DOS là một hệ điều hành bằng các dòng lệnh, nó yêu cầu bạn phải đưa vào các lệnh, các biến, và các cú pháp mới sử dụng thành công MS-DOS. Tuy nhiên, sau khi hiểu các lệnh MS-DOS, bạn có thể đạt được trình độ điều khiển cao hơn thông qua các khả năng của hệ điều hành - đó là việc lập các thuộc tính tệp, tạo ra các tệp bó thực hiện tự động, và xây dựng các thủ tục ghi dự phòng bán tự động. Cùng với khả năng này, bạn cũng có sẵn các trình tiện ích rất tốt giúp bạn tránh dùng DOS bằng cách kết hợp các nhiệm vụ đó trong một giao diện người-máy với các trình đơn kéo xuống và các hộp hội thoại. Một hạn chế ngặt nghèo nhất của MS-DOS là hàng rào 640 RAM mà hệ điều hành bắt buộc quá trình điện toán tương thích IBM-PC phải tuân theo. Mặc dù có nhiều người đã chuyển sang với Windows để sử dụng các khả năng quản lý bộ nhớ và giao diện dễ dùng của nó, nhưng hàng triệu máy tính tương thích IBM PC loại cũ đang tồn tại không thể chạy tốt với Windows. Không nghi ngờ gì nữa, MS-DOS là một hệ điều hành được dùng rộng rãi nhất trên thế giới và có lẽ sẽ còn tồn tại trong nhiều năm nữa.

Lời khuyên: Để có hiệu suất cao nhất với Microsoft Windows, bạn hãy nâng cấp lên dùng MS-DOS 6.0 - 6.2 và chạy trình tiện ích MEMMAKER để tận dụng hết ưu điểm của toàn bộ nhớ của hệ thống.

b. Microsoft Windows

Đây là một hệ điều hành phổ thông nhất. Windows là một môi trường cửa sổ và giao diện người - máy theo ứng dụng (API), nhằm bổ sung thêm các thao tác đa nhiệm cho DOS, và đưa vào quá trình điện toán theo quy cách IBM một số tính năng giao diện người - máy theo đồ hoạ của Macintosh, như các trình đơn kéo xuống, các kiểu chữ đa dạng, các dụng cụ văn phòng, và khả năng di chuyển tài liệu từ chương trình này sang chương trình khác thông qua Clipboard. Vì Windows có tất cả các chức năng cần thiết cho việc

bổ sung thêm các tính năng như các chương trình đơn, các cửa sổ, và các hộp hội thoại, cho nên tất cả các chương trình ứng dụng Windows đều có một giao diện trợ giúp. Ban đầu, Windows chỉ có nhiều hơn một ít so với bản duyệt trước của Presentation Manager của OS/2. Windows chỉ chạy được với một ít chương trình ứng dụng được biên soạn đặc biệt cho nó trong môi trường giao diện người-máy đồ hoạ, và sử dụng các tiêu chuẩn nhớ mở rộng như Lotus-Intel-Microsoft Specification 4.0 chẳng hạn. Tuy nhiên, những chương trình theo quy cách IBM thế hệ mới đòi hỏi phải có chế độ bảo vệ của các bộ vi xử lý Intel 80286 và 80386 với 16 M không gian nhớ đồng nhất của chúng. Microsoft đã chọn lập trường là các chương trình trong chế độ bảo vệ phải được biên soạn cho OS/2 và Presentation Manager. Về sau, DESKQ view (của hãng Quarterdeck Systems) xuất hiện với khả năng có thể chạy các chương trình MS-DOS trong các cửa sổ chế độ bảo vệ, cho nên Microsoft cũng đã cho ra đời Windows 3.0 có thể chạy với các chương trình ứng dụng MS-DOS trong chế độ bảo vệ. Windows 3.0 đã thành công một cách kỳ diệu, cùng với Excel và Word for Windows đang thu được nhiều người dùng hơn; các khách hàng sử dụng Windows khác thì mua chương trình này đơn giản chỉ vì muốn chạy nhiều chương trình cùng một lúc. Bản 1992 của Windows 3.1 đã củng cố thêm cho sự thông dụng của Windows, do có nhiều cải thiện quan trọng về tốc độ, có Program Manager và File Manager, có các phông True Type, có khả năng điều khiển chuột trong các trình ứng dụng MS-DOS, có màn hình Help được cải thiện và khả năng đa nhiệm, có các trình cất giữ màn hình cài sẵn, có khả năng liên kết và nhúng đối tượng (OLE), và các mở rộng đa phương tiện. Mặc dù nhiều người sử dụng DOS vẫn tiếp tục thích thú về tốc độ cao hơn và sự kiểm soát của người dùng trực tiếp hơn trong các trình ứng dụng được ưa thích của họ, nhưng những người sử dụng có năng khiếu đồ hoạ đều đồng ý Microsoft Windows sử dụng rất thú vị. Lời khuyên: Để chạy Windows, bạn phải có nhiều sức mạnh điện toán hơn khi chạy các chương trình ứng dụng DOS tương đương. Một nền Windows tối thiểu phải gồm có một máy 80386 SX chạy ở 16 MHz (như) g tốt hơn là 20 hoặc 25 MHz, ít nhất 4 M RAM, 80 M đĩa cứng, một bộ điều hợp VGA 16 bit cùng với màn hình VGA, và một máy in phun hoặc laser. Tốt nhất là mua một máy 80486 DX 2 chạy ở 66 MHz, 8 M RAM, 200 M đĩa cứng, và một bộ điều hợp VGA cùng với màn hình.

Sau Windows 3.1 đến các phiên bản Win 95, Win 98, Win 2000.

Microsoft Windows NT

Là phiên bản 32 bit của giao diện người-máy đồ hoạ thông dụng của Microsoft, nó tạo khả năng đa nhiệm thực sự cho những máy tính cá nhân có cơ sở Intel và các trạm công tác chuyên dụng. Phiên bản này của Windows có thể bỏ qua DOS và có khả năng thâm nhập tốt hơn vào bộ nhớ hệ thống so với Windows 3.1. Windows NT chạy được trong chế độ bảo vệ không loại trừ trường hợp nào, cho phép những lập trình viên có thể sử dụng đến 4 gigabyte RAM mà không cần phải sắp xếp lại để đánh lừa máy. Cùng với những ưu điểm đó, Windows NT còn có khả năng chấp nhận trục trặc, quản lý tệp, thâm nhập mạng, và bảo vệ an toàn được cải thiện tốt hơn. Trên màn hình, Windows NT trông giống như Windows 3.1 quen thuộc. Không lệ thuộc vào sự giới hạn của DOS, Windows NT nhằm vào những nhược điểm của Windows 3.1 để khắc phục, nên đã trở nên hấp dẫn đối với các hệ điện toán nhiều người sử dụng, là các hệ hay dùng UNIX hoặc OS/ 2. Tuy nhiên với đòi hỏi bộ nhớ lớn - 70 M không gian đĩa cứng và 16 M RAM chưa dùng đến - Microsoft Windows NT chỉ được sử dụng trong các hệ máy tính mạnh nhất.

c. Các phần mềm ứng dụng khác

Phần mềm ứng dụng hiện nay có cho mọi ngành nghề, mọi thành phần công việc và không ngừng được cải tiến.

Với công tác thiết kế cầu đường ô tô, đã có các phần mềm về thiết kế bình đồ, cắt dọc, cắt ngang đường *(Softdesk)*, tính kết cấu *(SAP)*, tính ổn định mái taluy *(Bishop)*, tính mạng lưới thoát nước đô thị *(Mouse)*, tính hiệu quả kinh tế vốn đầu tư *(CBA, HDM)* v.v...

Công nghệ phần mềm của Việt Nam hiện nay cũng được phát triển khá mạnh. Trong thiết kế đường, nhiều kỹ sư đã quen thuộc với phần mềm NOVA của Công ty Hài Hoà.

Tuy nhiên, phạm vi ứng dụng phần mềm đôi khi cũng không thích hợp do cơ sở lý thuyết khác nhau, quy trình kỹ thuật khác nhau, và không phải lúc nào cũng *"có sẵn"*, *"có đủ"* để cài đặt vào máy tính. Do vậy, các kỹ sư nên biết lập trình để có thể *"tự biên tự diễn"* một số nội dung thiết kế theo quy trình Việt Nam, theo lý thuyết tính toán mà mình cho là hợp lý.

3) Công tác lập trình

Để lập trình, trước hết phải dựa vào ngôn ngữ lập trình. Hiện nay có nhiều ngôn ngữ lập trình như Pascal, C+, Visual Basic ...

Để lập trình cho ra một sản phẩm phần mềm, thí dụ thiết kế đường, yêu cầu đầu tiên là phải có sự kết hợp chặt chẽ của 2 chuyên gia:

- Chuyên gia về thiết kế đường, chịu trách nhiệm đề ra các nội dung thiết kế, trình tự thiết kế, các tình huống xảy ra khi thiết kế...

- Chuyên gia về lập trình máy tính, chuyển các nội dung trên thành chương trình chạy máy tính qua các bước nhập số liệu, các lệnh ...

Nếu chuyên gia thiết kế đường không giỏi, sẽ không thể lường trước mọi tình huống xảy ra để chuyên gia lập trình viết chương trình. Ngược lại, nếu chuyên gia lập trình không giỏi, không nắm được vấn đề để chuyển thành chương trình cho máy chạy. Kết quả là chương trình chạy sẽ bị lỗi, không đáp ứng hết mọi tình huống thực tế khi người kỹ sư thiết kế sử dụng phần mềm để thiết kế.

Chương 2

KHÁI QUÁT VỀ LẬP TRÌNH TURBO PASCAL

2-1. ĐẶC ĐIỂM NGÔN NGỮ PASCAL

1) Xuất xứ

Năm 1971, giáo sư Niklaus Wirth ở trường đại học kỹ thuật Zurich (Thuỵ sỹ) đã thiết kế và công bố một ngôn ngữ lập trình bậc cao, vạn năng, được đặt tên là Pascal để tưởng nhớ đến nhà toán học người Pháp Blaise Pascal ở thế kỷ XVII.

Trong lập trình Pascal, lần đầu tiên N. Wirth đưa ra khái niệm "***Phương pháp lập trình cấu trúc***" mà trước đó, như phương pháp FORTRAN không có. Ngày nay, Pascal được coi là một trong những ngôn ngữ lập trình bậc cao ưu việt nhất trong cả lĩnh vực giảng dạy và lập trình chuyên nghiệp.

2) Các loại Pascal

Ban đầu Pascal chỉ là ngôn ngữ để dạy học là chính, nhưng do khả năng ứng dụng thực tế, các hãng phần mềm đã sử dụng để thương mại hoá với nhiều loại khác nhau:

- ISO Pascal: Pascal chuẩn ISO (International Standard Organization).
- ANSI Pascal: Pascal của Viện tiêu chuẩn Hoa Kỳ (American National Standard Institute).
- Turbo Pascal: Pascal của hãng Borland.
- IBM Pascal: Pascal của hãng Microsoft.
- UCSD Pascal: Pascal của trường Đại học tổng hợp California ở San Diego (University of California at San Diego).
- Pascal chạy trên máy tính VAX

Trong các loại trên, Turbo Pascal là loại phổ biến nhất trên thế giới, vì đáp ứng những yêu cầu rất đa dạng của người sử dụng.

- *Đối với người học*: Turbo Pascal cho phép người học làm việc với máy theo chế độ đối thoại (question – answer).

- *Đối với người lập trình chuyên nghiệp*: Turbo Pascal là một công cụ phát triển mạnh và có tốc độ cao vì có thời gian dịch và thực hiện chương trình nhanh vào loại bậc nhất.

3) Các version của Turbo Pascal

Hãng Borland đã cho ra đời các Turbo Pascal (TP) version 2.0, 3.0, 4.0, 5.0, 5.5, 6.0 và 7.0.

Version 6.0: đặc điểm nổi bật là giới thiệu thành phần của Turbo vision, một thư viện chương trình cụ thể, định hình có các cửa sổ tiện ích để đối chiếu với nhau.

Version 7.0: được đưa ra vào cuối năm 1992, có thể chạy trong cả hệ điều hành MS-DOS và WINDOWS. Với version này, ngoài Turbo Pascal còn có Borland Pascal (BP). Version này đã thực sự mở rộng cú pháp, quản lý chuỗi 0, củng cố lại các quy tắc, hệ thống, chỉ thị các câu lệnh.

Sau khi cho ra đời version 7.0, hãng Borland đổi tên thành Inprise và để lại một tồn tại: khi chạy Borland Turbo Pascal 7.0 trên máy tính tốc độ cao (Pentium II, AMD K6...) thường mắc lỗi đưa ra thông báo "*Erro 200: division by zero*", ngay cả khi chương trình không hề có phép chia. Người dùng trên thế giới đã phải tự khắc phục thiếu sót này. Ở Việt Nam, bạn đọc có thể tự sửa bằng cách copy file TURBO.TPL từ Pascal đã chạy được trong một máy tốc độ cao, sau đó copy đè lên file Turbo.tpl đã có trong thư mục BIN của máy chưa chạy được.

4) Đặc điểm của ngôn ngữ

Ngôn ngữ Pascal có các đặc điểm sau:

- *Là ngôn ngữ có tính định kiểu chặt chẽ.*

Các biến và các hằng trong một biểu thức phải có cùng kiểu dữ liệu. Khi có kiểu khác nhau thì phải chuyển đổi về cùng kiểu. Có thể chuyển đổi tự động hoặc chuyển đổi bắt buộc.

- Là ngôn ngữ mang tính cấu trúc.

Pascal cho phép một chương trình lớn có thể chia thành nhiều phần, mỗi phần độc lập với nhau nên nhiều người có thể cùng tham gia viết một chương trình. Thông thường các khối ứng với từng nhiệm vụ cụ thể được thực hiện bằng các chương trình con (subroutine) và sử dụng các biến riêng (local variable) nên không làm ảnh hưởng đến các chương trình con của các khối khác. Quan hệ giữa chương trình con với chương trình chính hay giữa chương trình con với nhau là xử lý dữ liệu và đưa ra kết quả từ chương trình con đó.

Cụ thể, tính cấu trúc của Pascal được thể hiện ở 3 mặt sau:

+ *Cấu trúc về mặt dữ liệu*: từ các dữ liệu chuẩn hay dữ liệu cấu trúc, người lập trình có thể xây dựng các kiểu dữ liệu phức tạp hơn.

+ *Cấu trúc về mặt lệnh*: từ các lệnh đơn hay lệnh cấu trúc, người lập trình có thể nhóm chúng lại với nhau và đặt giữa từ khoá BEGIN...END thành câu lệnh phức tạp hơn gọi là *lệnh phức* hay câu *lệnh ghép*.

+ *Cấu trúc về mặt chương trình*: một chương trình có thể chia thành nhiều chương trình con dưới dạng các *thủ tục* (procedure), *các hàm* (funtions), các *đơn thể* (modules), các *đơn vị* (units) độc lập với chương trình chính mà sự tương tác giữa chúng là dự liệu đưa vào và nhận lại kết quả.

Khái niệm về lập trình cấu trúc sẽ giới thiệu chi tiết hơn ở mục 2-7.

2-2. KHỞI ĐỘNG, GHI CHƯƠNG TRÌNH VÀ KẾT THÚC TURBO PASCAL

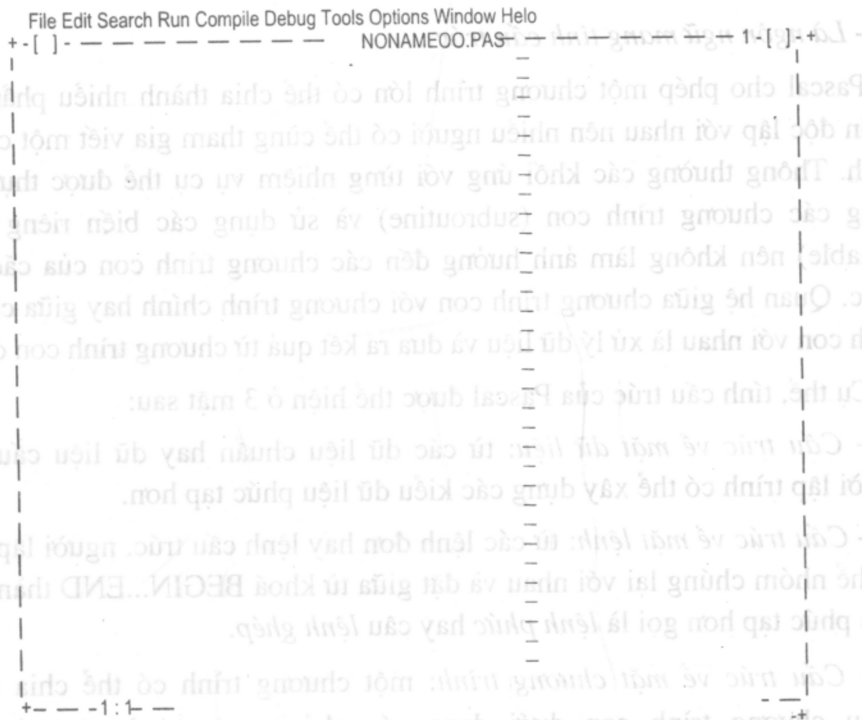
1) Khởi động Turbo Pascal 7.0 (TP7), hoặc Borland Pascal (BP).

Để làm việc với Borland-Turbo Pascal ta nên đặt tất cả các tệp hệ thống trong một thư mục (như TP7, BP) ở ổ C. Để khởi động TP7 hoặc BP qua NC (Norton Commande), ta vào NC, lần lượt đặt thanh sáng rồi enter: TP – BIN – turbo.exe - sẽ ra file đầu tiên với menu chính như hình 2-1.

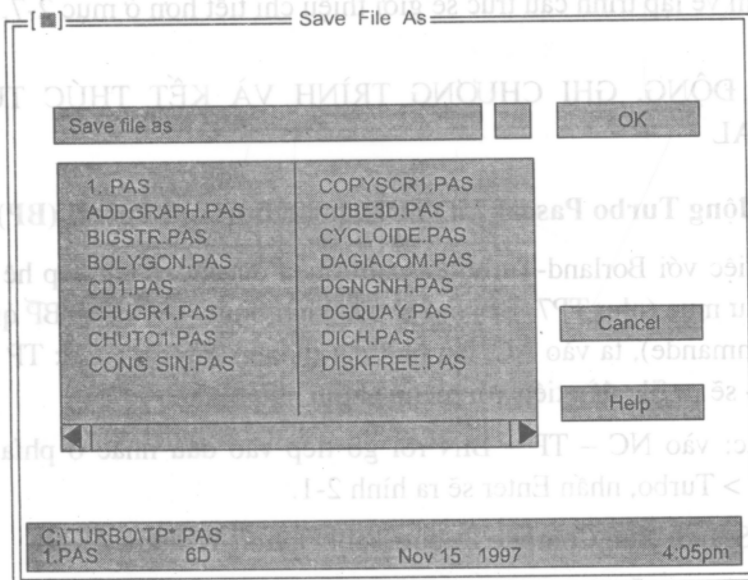
Cách khác: vào NC – TP – BIN rồi gõ tiếp vào dấu nhắc ở phía dưới: C:\TP > BIN > Turbo, nhấn Enter sẽ ra hình 2-1.

File Edit Search Run Compile Debug Tool Option Window Help

Như lẽ thường, mỗi menu trên sẽ bao gồm nhiều mục chọn khác, qua các hộp thoại mà chúng ta sẽ lần lượt tìm hiểu sau.



Hình 2-1: Màn hình File mới của TP7.



Hình 2-2: Hộp thoại ghi tên file mới (Save File As).

2) Ghi tên File (tệp chương trình)

Sau khi soạn thảo chương trình lần đầu, cần ghi lại và đặt tên File mới (Save File As) ta nhấn F2 ra hộp thoại như hình 2-2. Trong ô đầu mục *Save file as* ta ghi tên file với đuôi * .PAS (thí dụ: *phepcong.PAS*).

3) Ra khỏi Turbo Pascal

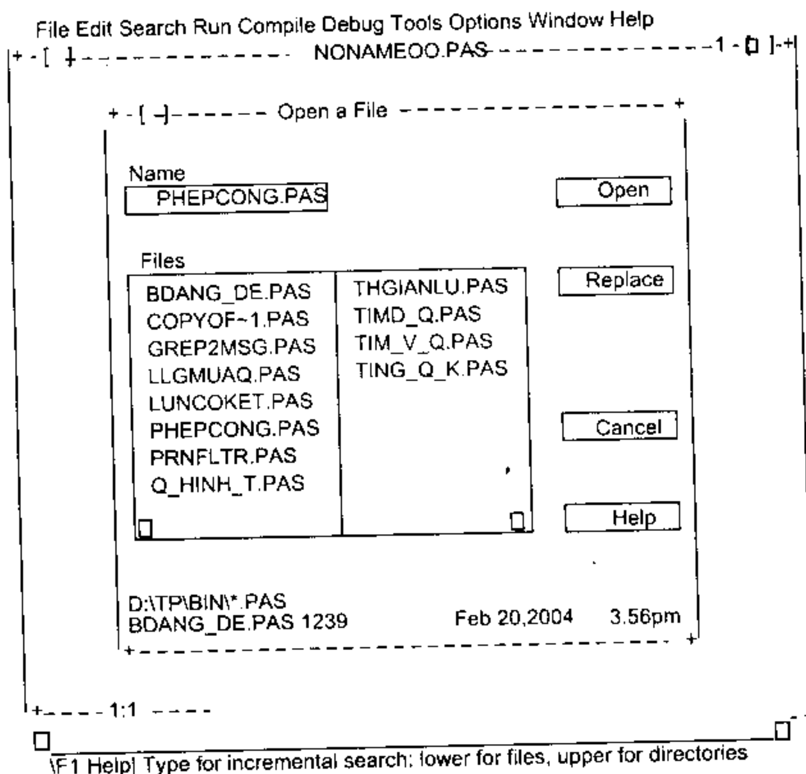
Cách 1: dùng chuột chỏ vào menu File, chọn Exit.

Cách 2: nhấn Alt+X

4) Mở lại File đã có

Cách 1: dùng chuột đưa con chỏ vào menu File, chọn Open ra hộp thoại *Open a File* như hình 2-3. Nhấn tiếp chuột vào File đã có định mở lại.

Cách 2: nhấn F3 ra hộp thoại *Open a File* như hình 2-3., Enter đưa thanh sáng xuống danh sách File, chuyển thanh sáng đến File đã cố định mở lại, Enter.



Hình 2-3: Hộp thoại Open a File

5) In chương trình và kết quả chạy chương trình

a. In chương trình

Sau khi soạn thảo xong chương trình, chạy đúng, muốn in lại, có thể làm như sau:

- Nhấn Alt+X ra khỏi TP

- Vào Word – Open – ổ C: - TP – BIN – Tìm tên File (thí dụ Pro-4.PAS) – Open – Print – OK.

Chú ý: trong BIN sẽ có nhiều File Pro-4 với các đuôi như exe, bak...Máy chỉ in với đuôi PAS.

b. In kết quả chạy chương trình

Sau khi chạy chương trình ra kết quả trên màn hình DOS, muốn in làm như sau:

- Nhấn Alt+Screen – nhấn enter về chương trình – nhấn Alt+X thoát khỏi TP

- Vào Word (ra màn hình trắng) – vào menu Edit chọn Paste (hiện ra kết quả chương trình trong Word) – Print – OK.

2-3. SOẠN THẢO VÀ THỰC HIỆN CHƯƠNG TRÌNH

1) Soạn thảo chương trình

Có thể hình dung khái quát một chương trình Pascal: Là một bài toán gồm các biến số (được thể hiện bằng các kiểu dữ liệu) trong một hàm số, được giải bằng các lệnh liên kết với nhau qua các từ khoá. Chi tiết sẽ lần lượt giới thiệu ở mục 2-4, 2-5, 2-6.

Trước khi soạn thảo, khởi động vào File mới như đã nêu ở trên.

Khi diễn giải soạn thảo chương trình, ta có thể gõ tiếng Việt không dấu (tức là theo Font tiếng Anh).

Thủ tục soạn thảo một chương trình Pascal điển hình thường có dạng như ví dụ sau:

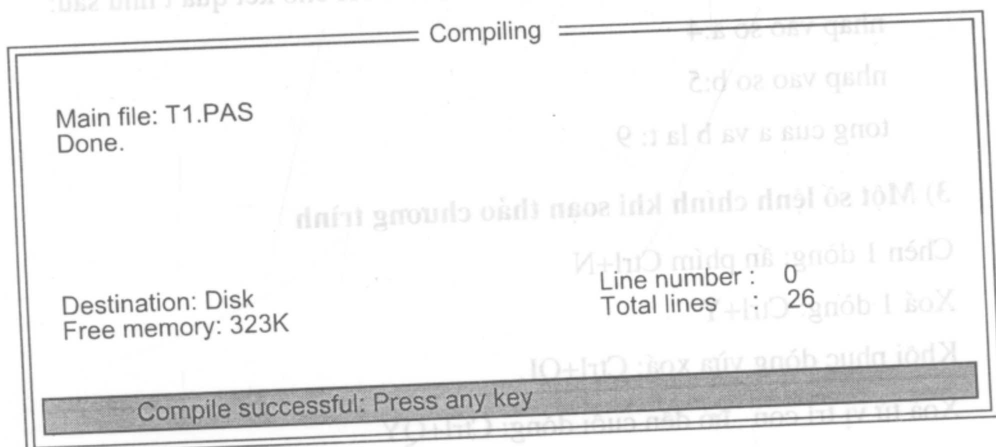
Ví dụ 2-1.

```
            Program Phep_cong;
uses crt;
var
        a,b,t:integer;
begin
 clrscr;
  write('nhap vao so a:');readln(a);
  write('nhap vao so b:');readln(b);
  t:=a+b;
  writeln('tong cua a va b la t:',t:10);
 readln
end.
```

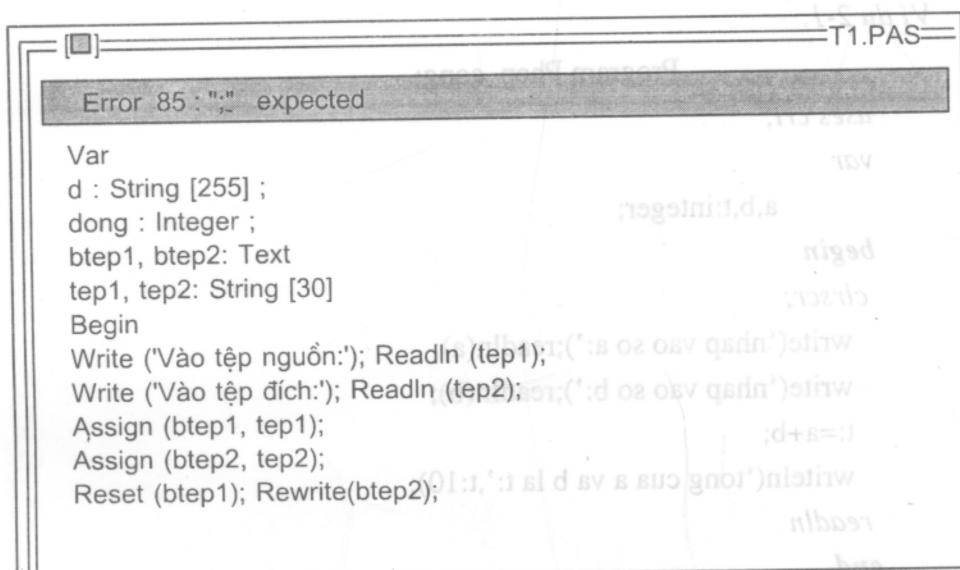
Để kiểm tra chương trình soạn thảo có đúng luật không, nhấn phím Alt+F9. Nếu đúng sẽ ra thông báo như hình 2-4. Nếu không đúng sẽ đưa ra thông báo như hình 2-5.

Chi tiết các thuật ngữ dùng trong ví dụ 2-1, sẽ giải thích ở mục 2-4, 2-5.

Số 10 sau chữ t có ý nghĩa là 10 khoảng trống dành chỗ cho viết số, gồm cả phần nguyên và thập phân (nếu là real).



Hình 2-4: *Thông báo kết quả kiểm tra lập chương trình đúng*



```
T1.PAS
Error 85 : ";" expected
Var
d : String [255] ;
dong : Integer ;
btep1, btep2: Text
tep1, tep2: String [30]
Begin
Write ('Vào tệp nguồn:'); Readln (tep1);
Write ('Vào tệp đích:'); Readln (tep2);
Assign (btep1, tep1);
Assign (btep2, tep2);
Reset (btep1); Rewrite(btep2);
```

Hình 2-5: *Thông báo lỗi khi chương trình viết sai*

2) Thực hiện (chạy) chương trình

Có 2 cách:

- Vào menu Run, chọn mục Run

- Nhấn phím Ctrl+F9

Sẽ ra màn hình nhắc nhập lần lượt số a, số b rồi cho kết quả t như sau:

nhap vao so a:4

nhap vao so b:5

tong cua a va b la t: 9

3) Một số lệnh chính khi soạn thảo chương trình

Chèn 1 dòng: ấn phím Ctrl+N

Xoá 1 dòng: Ctrl+Y

Khôi phục dòng vừa xoá: Ctrl+QL

Xoá từ vị trí con trỏ đến cuối dòng: Ctrl+QY

Đánh dấu khối: đưa con trỏ đến đầu vùng cần đánh dấu khối ấn Ctrl+KB, đưa con trỏ đến cuối khối ấn Ctrl+KK

Sao chép khối: Sau khi đã đánh dấu khối, khối được xác định sẽ đổi màu. Để sao chép khối này, đưa con chỏ đến vị trí cần sao chép ấn Ctrl+C

Dịch chuyển khối: Ctrl+V

Xoá khối: Ctrl+Y

Xoá đánh dấu khối Ctrl+KH

Tìm kiếm chuỗi ký tự: Ctrl+QF

Tìm kiếm và thay thế một chuỗi ký tự bằng một chuỗi ký tự khác: Ctrl+QA

2-4. CÁC KIỂU DỮ LIỆU

Kiểu dữ liệu dùng để xác định tập hợp các giá trị mà một biến có thể nhận. Mỗi biến trong chương trình cần phải được kết hợp với một và chỉ một kiểu dữ liệu. Kiểu dữ liệu xác định thuộc tính của dữ liệu, là cơ sở để phân biệt đối tượng này với đối tượng khác.

Các kiểu dữ liệu chuẩn của Pascal như bảng 2-1 sau.

Bảng 2-1. Bảng kê các dữ liệu chuẩn của Pascal

Tên kiểu	Kích thước	Phạm vi	Ghi chú
Integer	2 Byte	-32768 đến 32767	Dạng 8 bit, có dấu
Byte	1 Byte	0 đến 255	Dạng 8 bit, không dấu
Word	2 Byte	0 đến 65535	Dạng 16 bit, không dấu
ShortInt	1 Byte	-128 đến 127	Dạng 8 bit, có dấu
LongInt	4 Byte	-2147483648 đến 2147483647	Dạng 32 bit, có dấu
Boolean	1 Byte	2 giá trị: True hoặc False	
Char	1 Byte	0 đến 255	Dạng 8 bit, không dấu
Real	6 Byte	2.9E-39 đến 1.7E+38	Định trị có 11 ký số
Extended	10 Byte	3.4E-4932 đến 1.1E+4932	

Sau đây mô tả một vài kiểu dữ liệu chính:

1) Kiểu số nguyên Integer

Là các số nguyên như 0,1,3, 34, 57 .... có dấu âm (-) hoặc dương (+)

Cũng có thể là các chữ đại số như: a, b, m, n, x, y, so ...

2) Kiểu số thực Real

Là các số thập phân, có dấu âm (-) hoặc dương (+)

Các số thực được viết dưới 2 dạng:

+ Dạng dấy phẩy tĩnh như 654.326

+ Dạng dấu phẩy động như 6.54326E+02 ($=6.54326*10^2 = 654.326$)

3) Kiểu Boolean

Dữ liệu kiểu này chỉ có 2 giá trị *True* (đúng) và *False* (sai). Giá trị False được coi là nhỏ hơn True

2-5. TÊN BIẾN VÀ HẰNG, PHÉP GÁN VÀ SO SÁNH, LỜI CHÚ THÍCH, TỪKHOÁ

1) Tên

Tên dùng để đặt cho các đại lượng trong chương trình như tên biến, tên hằng, tên kiểu dữ liệu, tên hàm, tên mảng, tên bản ghi, tên chương trình, tên con trỏ... Tên bắt đầu phải là chữ cái và không chứa dấu trống, dấu phép toán và tên không được trùng với từ khoá.

Các ví dụ đúng về tên:

Tinh_luu_luong_cong_Q;

Delta;

Các ví dụ sai về tên:

Tinh-do-lun-nen;	(vì có dấu trừ)
2duong_thang;	(vì ký tự đầu tiên là số)
Type;	(trùng với từ khoá)

Trong khi thực hiện các câu lệnh, Pascal không phân biệt chữ hoa và chữ thường, (nhưng với chuỗi ký tự thì có phân biệt).

Ví dụ về tên sau đây đều có ý nghĩa như nhau:

TinhKhoiluong;

tinHkhOiLuonG;

2) Biến

Biến là một đại lượng lưu trữ một giá trị nào đó dùng để tính toán, so sánh trong chương trình. Biến cũng nhận các kiểu dữ liệu như đã giới thiệu ở trên và có thể thay đổi giá trị nhiều lần trong chương trình.

Biến là tên một vùng lưu trữ dữ liệu.

Trước khi dùng biến phải khai báo theo dạng sau:

var

<dãy các biến>: <dạng dữ liệu>

Các biến trong dãy được viết cách nhau một dấu phẩy, tham khảo ví dụ sau:

```
var
    a,b,c:Real;
    i,n:Integer;
    t:String[10];
```

Một biến dạng string được cấp một số byte bằng chiều dài của nó cộng thêm một. Byte đầu tiên để ghi số ký tự đang lưu giữ, mỗi byte còn lại một ký tự.

3) Hằng

Hằng là một đại lượng xác định, không thay đổi giá trị trong chương trình với từ khoá CONST. Hằng cũng có các kiểu dữ liệu như trên. Khi khai báo hằng, phải tuân thủ nguyên tắc sau:

Tên hằng = giá trị của hằng hoặc

Tên hằng = biểu thức của hằng

Ví dụ:

```
Const
    a:=be rong nen;
    b:= 60;
    c:=4*6
    d:=FALSE;
```

Turbo Pascal cho phép thực hiện đồng thời việc khai báo và khởi đầu một giá trị sau từ khoá const.

Ví dụ:

```
Const
    a:=10.5;
    b:integer=60;
```

Sau này, trong chương trình, giá trị của biến b đã được nhận giá trị là 60 (trừ trường hợp ta gán giá trị khác).

Một số lưu ý:

Trong chương trình có thể gán lại giá trị của biến b:

```
b:=48;
```

Nhưng nếu gán lại các hằng không được khai báo như dạng trên, (ví dụ như biến a) thì sẽ bị thông báo lỗi. Ví dụ như sau là không hợp lệ:

```
a:=8.4;
```

4) Phép gán

Trong khi lập chương trình, một số biến có thể được gán một giá trị nào đó bằng toán tử gán (:=). Phép gán còn có ý nghĩa là thay thế giá trị hiện tại của biến bằng một giá trị mới.

Ví dụ: a:=5; nghĩa là biến a nhận giá trị là 5

b:=3*h; nghĩa là biến b luôn có giá trị gấp 3 lần biến h

c:=c+1; nghĩa là biến c lấy giá trị cũ tăng thêm 1

Cần lưu ý rằng: biểu thức được gán phải phù hợp với kiểu dữ liệu của biến hoặc kết quả của hàm.

Ví dụ:

Nếu khai báo	*Phép gán hợp lệ*	*Phép gán không hợp lệ*
var		
i:Integer;	i:=18;	i:=18.5;
k:Real;	k:=10.6;	k:=10;
h:Char;	h:='C';	h:=5;
t:String;	t:='Mai Anh';	t:=Mai Anh;

Các phép gán sau đây cũng là không hợp lệ:

a-b:=8;

12:=c;

a-10:=b;

vì không biết viết vế trái vào đâu.

5) Phép so sánh logic

Phép so sánh logic được thực hiện bằng các ký hiệu

=	bằng nhau
>	lớn hơn
<	nhỏ hơn
>=	lớn hơn hoặc bằng
<=	nhỏ hơn hoặc bằng
<>	khác nhau

Để kết hợp các phép so sánh, Turbo Pascal dùng từ khoá OR để chỉ phép "hoặc" logic và dùng từ khóa AND để kết hợp 2 điều kiện.

Ngoài ra ta có thể dùng từ khoá NOT để chỉ phép phủ định.

Ví dụ:

if (a>b) OR (c>d) then

6) Lời chú thích

Trong chương trình, khi cần thêm các dòng chú thích để dễ hiểu, hoặc khi cần sửa đổi bổ sung, ta đưa các lời chú thích vào giữa 2 ký hiệu (*..*) hoặc {..}.

Cặp ngoặc [..] trong TP được dùng để khai báo mảng.

Ta có thể dùng nhiều cặp ngoặc (..) cho diễn giải công thức tính toán. Các phép toán và so sánh sẽ được thực hiện từ trong ra ngoài.

7) Từ khoá

Trong chương trình có các từ khoá mang nội dung về mặt ngữ nghĩa hoàn toàn xác định và được viết bằng chữ in hoặc chữ thường, nhưng phải cách các ký tự khác. Các từ khoá thường dùng là:

AND, ARRAY, BEGIN, CONST, DIV, DOWNTO, ELSE, END, FILE, FOR, FORWARD, FUNTION, GOTO, IF, IN, LABEL, MOD, NIL, NOT, OF, OR, PROCEDURE, PROGRAM, RECORD, REPEAD, SET, THEN, TO, TYPE, UNTIL, VAR, WHILE, WITH...

Trong các từ khoá trên, có các từ khoá luôn đi cùng nhau để tạo thành một cặp trong chương trình như BEGIN...END, REPEAD...UNTIL. Điều này nhắc người lập chương trình khi viết từ khoá mở đầu thì cần nhớ đến nơi cần từ khoá kết thúc.

8) Tên chương trình

Một chương trình có thể được đặt tên sau từ khoá Program như

Program Tinh_luu_luong_Q;

Nguyên tắc viết tên như đã nêu ở mục 1 trên.

2-6. LỆNH WRITELN, WRITE VÀ READLN, READ. USES CRT VÀ CLRSCR

1) Lệnh Writeln(...);

Lệnh dùng để in các giá trị chủ yếu mà ta cần biết trên một dòng màn hình, bắt đầu từ vị trí hiện thời của con trỏ, sau đó đưa con trỏ về dầu dòng tiếp theo.

Cú pháp sử dụng lệnh này là:

Writeln(biểu thức 1, biểu thức 2,..., biểu thức n);

Qua ví dụ 2-1 có một biểu thức t như:

Writeln('Tong cua a va b la t:',t:10);

Các biểu thức là đối số của lệnh Writeln(...) có thể là số nguyên, số thực, chuỗi ký tự (String), giá trị logic... và có thể kết hợp giữa chúng.

Ví dụ 2-2:

```
Var
    a,b:String[20];
```

```
Begin
   a:='Mai Anh - ';
   b:='Giáo viên';
  Writeln(a,b);
  Readln
End.
```

Khi chạy chương trình sẽ có kết quả hiện trên màn hình:

Mai Anh - Giáo viên

Nhưng nếu thay dòng lệnh Writeln(a,b) bằng 2 dòng:

```
Writeln(a);
Writeln(b);
```

thì kết quả nhận được là 2 dòng như sau:

Mai Anh -

Giáo viên

Ví dụ 2-3.

```
Var
   a:String[30];
   c,d:Integer;
Begin
   a:='Kết quả của phép so sánh:';
   c:=8;
   d:=10;
  Writeln(a,c:3,'>',d:3,'là:',c>d);
  Readln
 End.
```

Khi chạy nhận được kết quả là:

Kết quả của phép so sánh 8>10 là: FALSE

Lưu ý: số [30] sau String là chỉ khoảng trống để viết chuỗi ký tự 'kết quả của phép so sánh'. Khi không để đủ khoảng trống này, kết quả dòng chữ in ra sẽ bị thiếu.

2) Lệnh Write(...), Writeln;

Cú pháp lệnh Write(...):

Write(bt1,bt2,...btn);

Tương tự như lệnh Writeln(...), nhưng khi đưa kết quả ra màn hình, con trỏ không xuống dòng dưới mà xuất hiện ở vị trí tiếp theo trên cùng một dòng với kết quả.

Cú pháp lệnh Writeln;

Writeln;

Có điểm khác lệnh Writeln(...) là không đưa ra tham số trong dấu ngoặc, chỉ có tác dụng đưa con trỏ xuống đầu dòng tiếp theo, tạo khoảng cách trình bày bản in kết quả.

Tác dụng của lệnh Write tham khảo kết quả ví dụ 2-1.

3) Lệnh Readln(...); Read;

a. Lệnh Readln(...);

Lệnh Readln(...) thường viết sau lệnh Write(...) để nhập số liệu từng biến theo từng dòng, (xem ví dụ 2-1). Khi nhập xong, con trỏ nhảy xuống dòng dưới để nhập tiếp.

b. Lệnh Read(b1,b2,...,bn);

Lệnh này tương tự lệnh Readln(...) nhưng khi nhập xong giá trị cho các biến, con trỏ màn hình không chuyển xuống dòng dưới.

c. Lệnh Readln;

Lệnh Readln (không có tham số trong ngoặc) dùng để dừng chương trình tại một thời điểm nào đó để kiểm tra biến, kiểm tra kết quả...Muốn chương trình tiếp tục, nhấn phím Enter.

4) Kết hợp lệnh Writeln (hoặc Write) với Readln (hoặc Read)

Lệnh Writeln (hoặc Write) thường dùng để giới thiệu loại dữ liệu, sau đó đến lệnh Readln (hoặc Read) để ghi giá trị số liệu lên màn hình. Tham khảo các ví dụ đã nêu ở trên sẽ thấy rõ điều này.

Trong một câu lệnh có thể nhập 1 biến (như ví dụ 2-1, Write('nhap vao so a:');) hoặc nhiều biến (như ví dụ 2-2, Writeln(a,b);). Đọc nhiều biến xem thêm ví dụ 2-4.

Ví dụ 2-4. Program Diem_trung_binh;

```
Var
    t,l,h,tb:real;
    ht:String[20];
Begin
    Write('Ho va ten hoc sinh:');Readln(ht);
    Write('Nhap diem Toan, Ly, Hoa:');Readln(t,l,h);
    tb:=(t+l+h)/3;
    Writeln('Diem trung binh cua hoc sinh ',ht);
    Writeln(tb:5:2);
  Readln
End.
```

Chú ý: Khi nhập giá trị cho t, l, h cần phân biệt các biến này bằng 2 cách:

- Nhập xong một biến (như t) nhấn phím Enter để xuống dòng, sau đó nhập biến khác (như l).

- Nhập xong trị số t, nhấn phím cách rồi nhập tiếp l, lại nhấn phím cách nhập h.

5) Lệnh Uses crt; Clrscr;

Lệnh Uses crt; được dùng cùng với lệnh Clrscr; để xoá màn hình sau khi chạy xong một lần chương trình.

6) In có định dạng

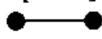
- Số nguyên

Giả sử ta có thông số n là số nguyên, muốn đưa giá trị ra màn hình với độ dài 5 vị trí, ta dùng lệnh:

```
Writeln('n=',n:5);
```

Giả sử n=243 kết quả thể hiện như sau:

Kết quả: [243]



5 vị trí

- Số thực

Nếu biến x là số thực, phải khai báo phần thập phân cho biến này bằng lệnh:

```
Writeln('x=',x:10:2);
```

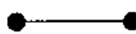
Kết quả được đưa ra trên màn hình dài 10 vị trí, trong đó có 2 vị trí số thập phân.

Giả sử x=2.14 kết quả đưa ra màn hình như sau:

Kết quả: [2.14]



2 vị trí số thập phân



10 vị trí

Ngoài ra Turbo Pascal còn cho phép ta đưa giá trị ra tại một vị trí nào đó. Trong trường hợp này, trước khi sử dụng lệnh Writeln, ta dùng lệnh GOTOXY theo cú pháp sau:

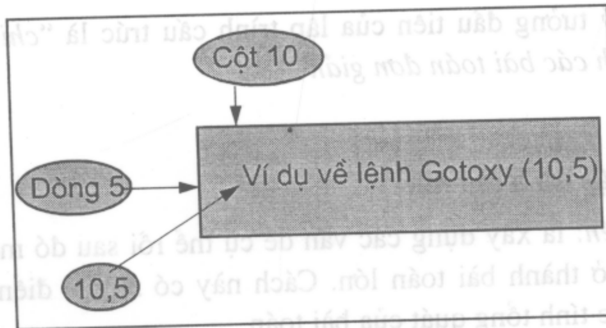
```
Gotoxy(x,y)
Writeln(<giá trị>);
```

Khi này <giá trị> được đưa ra tại toạ độ có cột là x và có dòng là y. Để thực hiện việc này, ta phải sử dụng Unit Crt như sau.

Ví dụ 2-5.

```
Uses crt;
Begin
    Goto(10,5);
    Writeln('Ví dụ về lệnh goto');Readln;
End.
```

Kết quả nhận được như hình 2-6.



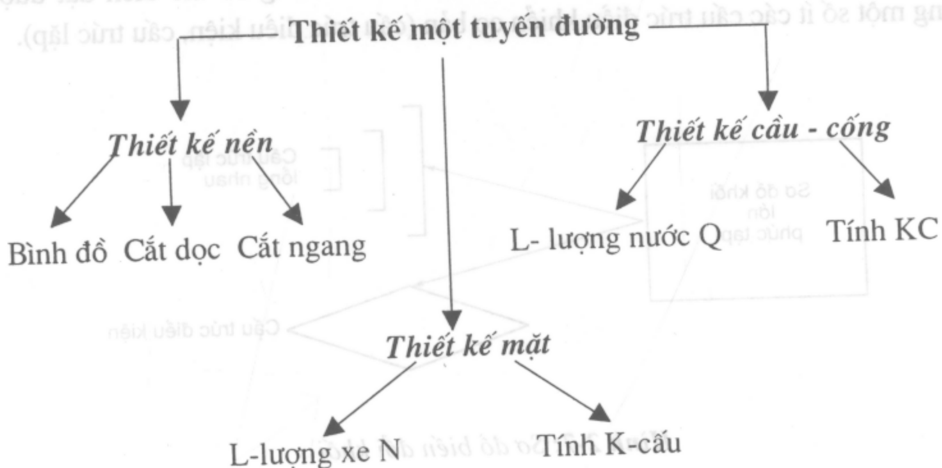
Hình 2-6: Kết quả nhận được khi dùng lệnh Gotoxy

2-7. KHÁI NIỆM VỀ LẬP TRÌNH CẤU TRÚC

Lập trình cấu trúc là một khái niệm rất khó định nghĩa nhưng có thể mô phỏng tóm tắt qua một số ý tưởng chính như sau:

1) Ý tưởng chính về lập trình cấu trúc

a. Về tổ chức chương trình: Yêu cầu chung là tổ chức phân cấp chương trình rõ ràng, có mối tương tác tối thiểu. Ví dụ:



Nguyên tắc chung của hệ phân cấp là cấp cao hơn là hệ thống của cấp thấp hơn. Phân cấp tới cấp thấp nhất không thể phân chia được nữa thì gọi là thành phần cơ bản.

Mỗi thành phần trong hệ nên thiết lập mối tương tác tối thiểu lên các thành phần khác sao cho rõ ràng và có thể kiểm soát được.

Như vậy, ý tưởng đầu tiên của lập trình cấu trúc là "*chia một bài toán phức tạp thành các bài toán đơn giản*".

b. Trình tự

Có 2 cách lập trình cấu trúc:

- *Từ dưới lên*: là xây dựng các vấn đề cụ thể rồi sau đó mới ghép nối lại với nhau để trở thành bài toán lớn. Cách này có nhược điểm là không thể hình dung được tính tổng quát của bài toán.

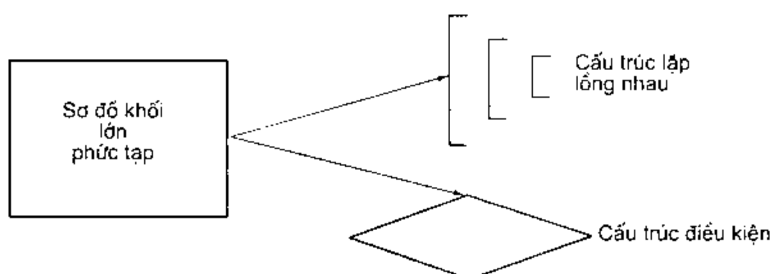
- *Từ trên xuống*: là cách phân tích tổng quát bài toán rồi sau đó mới giải quyết từng vấn đề cụ thể. Cách này thường dùng nhiều hơn.

c. Không cần lệnh Goto

Lập trình cấu trúc là lập trình không cần đến câu lệnh Goto để rẽ nhánh chương trình. Vì câu lệnh này làm cho chương trình rườm rà, khó hiểu, khó đọc, khó kiểm soát.

d. Sự biến đổi

Lập trình cấu trúc cho phép biến đổi bài toán có sơ đồ khối lớn, phức tạp thành những sơ đồ khối nhỏ dạng chuẩn sao cho chúng có thể diễn đạt được bằng một số ít các cấu trúc điều khiển cơ bản (cấu trúc điều kiện, cấu trúc lặp).



Hình 2-7: *Sơ đồ biến đổi khối*

e. Chương trình rõ ràng, sáng sủa

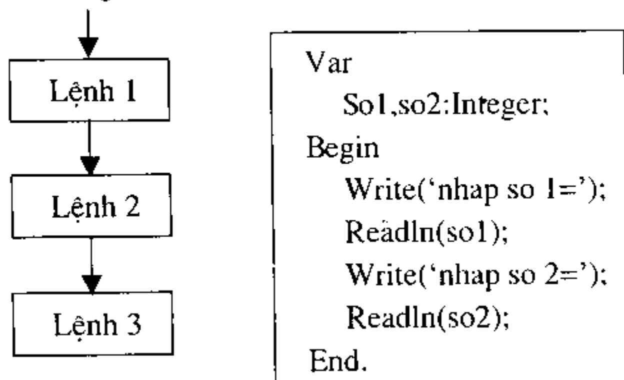
Lập trình cấu trúc là lập trình theo lối tuần tự từ trên xuống, kết hợp với các chú thích nên dễ dàng kiểm tra tính đúng đắn của chương trình. Do vậy, khi viết một chương trình, bản thân nó đã toát lên những câu hỏi và trả lời: "Làm gì ? Làm như thế nào ? Tại sao ? ".

2) Các cấu trúc điều khiển chuẩn

Trong lập trình cấu trúc, tuỳ theo chức năng nhiệm vụ người lập trình phải sử dụng các cấu trúc điều khiển sau đây:

a. Cấu trúc tuần tự (sequence)

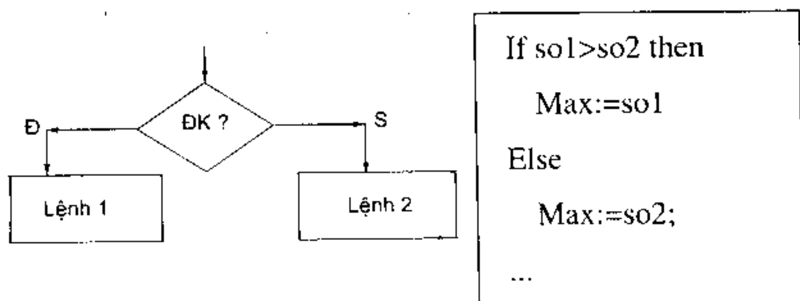
Trong cấu trúc tuần tự, các câu lệnh được sắp xếp và thực hiện theo thứ tự từ trên xuống dưới, tham khảo sơ đồ sau:



Hình 2.8: *Sơ đồ cấu trúc tuần tự*

b. Cấu trúc điều kiện (if)

Điều kiện là một biểu thức logic, có trị là kiểu Boolean, có nghĩa là chỉ có hai giá trị True (đúng) và False (sai). Nếu điều kiện đúng thì thực hiện lệnh 1, sai thì thực hiện lệnh 2. Tham khảo sơ đồ hình 2-9.



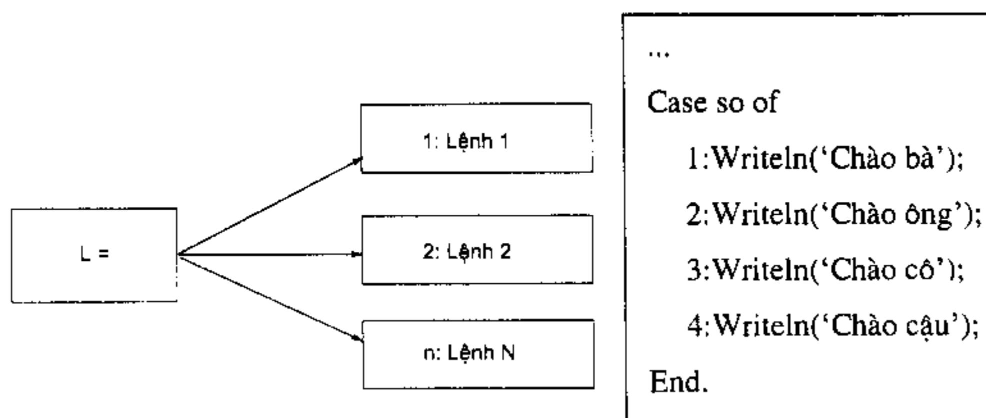
Hình 2-9: *Sơ đồ cấu trúc điều kiện*

Chú ý cách viết: trước từ khoá Else không yêu cầu có dấu (;)

c. Cấu trúc lựa chọn (case)

Cấu trúc lựa chọn khác với cấu trúc điều kiện là điều kiện để lựa chọn là một biến hoặc một biểu thức có giá trị xác định chứ không phải là giá trị

logic. Vì vậy nếu biến hoặc biểu thức có giá trị nào đó thì sẽ thực hiện câu lệnh tương ứng. Sơ đồ khối cấu trúc lựa chọn như hình 2-10.



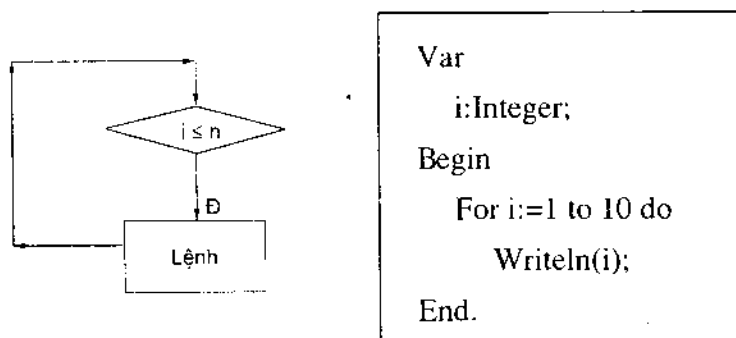
Hình 2-10: *Sơ đồ cấu trúc lựa chọn.*

Chú ý: thực tế lập trình diễn giải bằng chữ Việt không dấu. Một số chương trình mẫu ở đây dùng chữ Việt có dấu chỉ để bạn đọc dễ hiểu.

d. Cấu trúc lặp (loop)

Cấu trúc lặp, còn gọi là chu trình, là khi điều kiện còn thoả mãn thì còn tiếp tục thực hiện lệnh, ngược lại thì thoát ra khỏi vòng lặp. Tuỳ theo điều kiện lặp người ta chia ra:

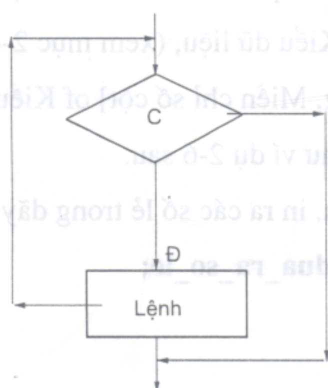
- *Số lần lặp xác định*: nếu trong một khối cần phải lặp đi lặp lại N lần một hoặc vài tác vụ nào đó thì sơ đồ cấu trúc như hình 2-11 sau:



Hình 2-11: *Sơ đồ số lần lặp biết trước*

- *Số lần lặp không xác định*: khi số lần lặp không được xác định thì tuỳ theo yêu cầu của tác vụ cần thực hiện mà chia ra 2 loại cấu trúc phụ như sau:

+ *Kiểm tra điều kiện trước*: Nếu điều kiện (C) đúng (Đ) thì thực hiện nhóm lệnh cho tới khi nào điều kiện sai (S) thì thoát khỏi cấu trúc. Sơ đồ khối của cấu trúc này như hình 2-12.

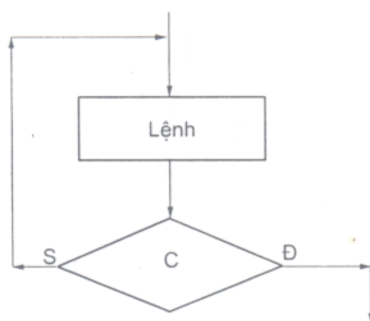



```
Var
    C:char;
Begin
    Write('Mật mã=');
    Readln(c);
    While c<>'N' do
    Write('Mật mã=');
    Readln(c);
End.
```

***Hình 2-12:** Sơ đồ số lần lặp không biết trước (kiểm tra điều kiện trước)*

Đoạn chương trình ví dụ này minh hoạ câu lệnh lặp While. Nếu ta gõ vào một ký tự khác với 'N' thì máy cứ bắt ta phải tiếp tục gõ cho đến khi đúng chữ 'N' mới thoát khỏi chương trình.

+ *Kiểm tra điều kiện sau*: Cấu trúc này được sử dụng khi cần thực hiện lệnh trước rồi mới kiểm tra điều kiện sau. Nếu điều kiện đúng (Đ) thì thoát ra khỏi chương trình. Xem sơ đồ hình 2-13.

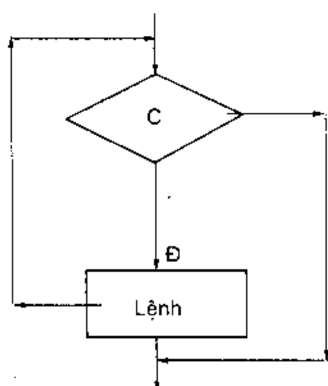



```
Var
    C:char;
Begin
    Repeat
        Write('Mật mã=');
        Readln(c);
    Until c='Y';
End.
```

***Hình 2-13:** Sơ đồ số lần lặp không biết trước (kiểm tra điều kiện sau)*

- *Số lần lặp không xác định*: khi số lần lặp không được xác định thì tuỳ theo yêu cầu của tác vụ cần thực hiện mà chia ra 2 loại cấu trúc phụ như sau:

+ *Kiểm tra điều kiện trước*: Nếu điều kiện (C) đúng (Đ) thì thực hiện nhóm lệnh cho tới khi nào điều kiện sai (S) thì thoát khỏi cấu trúc. Sơ đồ khối của cấu trúc này như hình 2-12.

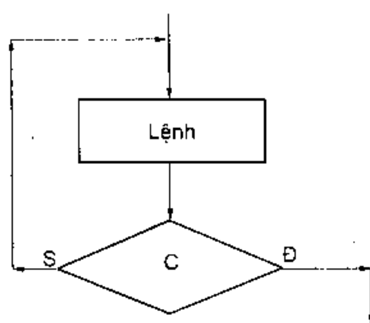



```
Var
   C:char;
Begin
   Write('Mật mã=');
   Readln(c);
   While c<>'N' do
   Write('Mật mã=');
   Readln(c);
End.
```

Hình 2-12: *Sơ đồ số lần lặp không biết trước (kiểm tra điều kiện trước)*

Đoạn chương trình ví dụ này minh hoạ câu lệnh lặp While. Nếu ta gõ vào một ký tự khác với 'N' thì máy cứ bắt ta phải tiếp tục gõ cho đến khi đúng chữ 'N' mới thoát khỏi chương trình.

+ *Kiểm tra điều kiện sau*: Cấu trúc này được sử dụng khi cần thực hiện lệnh trước rồi mới kiểm tra điều kiện sau. Nếu điều kiện đúng (Đ) thì thoát ra khỏi chương trình. Xem sơ đồ hình 2-13.




```
Var
   C:char;
Begin
   Repeat
      Write('Mật mã=');
      Readln(c);
   Until c='Y';
End.
```

Hình 2-13: *Sơ đồ số lần lặp không biết trước (kiểm tra điều kiện sau)*

Đoạn chương trình này yêu cầu gõ vào một ký tự cho tới khi đúng chữ (Y) mới thoát khỏi chương trình.

3) Cấu trúc mảng (Array)

Mảng là một dạng biến nhớ có thể chứa nhiều phần tử. Các phần tử mảng có cùng kiểu với nhau. Mỗi phần tử xác định qua 1 chỉ số. Có 2 loại mảng:

Mảng 1 chiều: Array [Miền chỉ số] of Kiểu dữ liệu, (xem mục 2-4).

Mảng 2 chiều: Array [Miền chỉ số hàng, Miền chỉ số cột] of Kiểu dữ liệu

Để dễ hiểu, ta cho chạy chương trình như ví dụ 2-6 sau.

Ví dụ 2-6: Nhập vào một dãy số nguyên, in ra các số lẻ trong dãy

```
          Program Mang_day_so_dua_ra_so_le;
uses crt;
var
        a: Array[1..50] of Integer;
        n, i: Integer;
begin
clrscr;
         write ('nhap vao so n:'); readln(n);
     for i:=1 to n do
        begin
         write('a[',i,']:='); readln(a[i]);
       end;
         write(Day da nhap:');
     for i:=1 to n do
         write (a[i]:5);
         writeln;
     for i:=1 to n do
  if a[i] mod 2<>0 then
         write (a[i]:5);
readln;
end.
```

Đặc điểm trong chương trình trên: có dùng phép chia lấy phần dư ***mod*** (như 5 *mod* 2 bằng 1) và phép chia lấy phần nguyên ***div*** (như 5 *div* 2 bằng 2).

n là số các số hạng ta nhập vào, thí dụ n = 6. Các số nhập vào là số chẵn, lẻ, âm, dương tuỳ ý. Kết quả chạy và in ra được như sau:

Nhap vao so n: 6

a[1]:= 4

a[2]:= -8

a[3]:= 7

a[4]:= -5

a[5]:= -2

a[6]:= 9

Day da nhap: 4 -8 7 -5 -2 9

7 -5 9

Trong đó đã đưa ra kết quả đúng là các số lẻ 7, -5, 9 .

4) Xâu ký tự (String)

Xâu ký tự dùng để chứa 1 dãy các ký tự, trong đó vừa có thể truy nhập đến từng phần tử ký tự như mảng 1 chiều, vừa có thể thao tác trên toàn bộ xâu như đọc, ghi ..

Khai báo: Var Tên xâu: String;

Tên xâu: String[n];

Các hàm thao tác trên xâu:

Delete(S,i,n); Xoá trong S từ vị trí i, n ký tự

Insert(p,S,i); Chèn vào xâu S từ vị trí i xâu con p

Val(S,a,i); Đổi xâu S ra số và cất vào biến a, nếu thành công i = 0;

Pos(p,S); Cho vị trí của xâu con trong xâu S, nếu không xuất hiện Pos = 0.

Ví dụ 2-7: Đếm chữ a trong một âu ký tự

Program Dem_chu_a_trong_xau_ky_tu;

uses crt;

```
var
        a: String;
        t,d,i: Integer;
begin
clrscr;
        write('Xau a:'); readln(a);
        writeln;
        t:=0;
        d:=0;
    for i:=1 to length(a) do
        if a[i] = 'a' then
          begin
            d:= d+1;
          end;
            writeln('So chu a la:', d);
readln
end.
```

Nhấn Ctrl+F9 cho chạy thử ra kết quả:

Xau a: phamhamaianh

So chu a la: 4

Ví dụ 2-8: Viết chương trình dòng chữ chạy.

```
Program Dong_chu_chay;
uses crt;
var
    a: String;
    x,y,n: Integer;
begin
clrscr;
        write('nhap vao xau ky tu a:'); readln(a);
clrscr;
        gotoxy(12,7);
```

```
            write(a);
            delay(800);
            x:=12;
    repeat
        clrscr;
            gotoxy(x+3,7);
            write(a);
            delay(800);
            x:=x+3;
        until x>=80;
    readln
    end.
```

Ý nghĩa các số liệu trên là: trên màn hình soạn thảo Pascal, theo khoảng cách con trỏ thì chiều ngang có 80, chiều dọc có 20. Toạ độ x (ngang), y (dọc) của vị trí được xác định bằng đơn vị này. Do đó, vị trí đầu tiên của dòng chữ là x = 12, y = 7. Để dòng chữ chạy ngang màn hình nên trị số y = 7 = const. Dòng chữ sẽ dịch chuyển mỗi lần 3 đơn vị nên có lệnh gotoxy(x+3,7). Lệnh until x>=80 có nghĩa là dòng chữ chỉ chuyển đến hết chiều ngang màn hình là dừng lại.

Delay800 là thời gian dừng sau mỗi nấc chạy.

5) Bản ghi (Record)

a. Đặc tính: Record là một cấu trúc bao gồm nhiều thành phần. Các thành phần có thể thuộc các kiểu dữ liệu khác nhau và được gọi là các trường (Field), mỗi trường đều được đặt tên.

b. Sử dụng Record: muốn truy nhập một biến Record, ta phải truy nhập theo thành phần của chúng. Cú pháp để đến một thành phần nào đó là:

<Tên biến Record>.<Tên trường>

c. Chú ý:

+ Không được dùng các thủ tục đọc và ghi: Read, Readln, Write, Writeln cho cả một biến kiểu Record.

+ Không được so sánh các Record bằng các phép toán quan hệ như: <, <=, >=, >. Riêng với các phép toán khác nhau (<>) và = thì có thể được dùng với 2 biến có cùng một kiểu Record.

+ Không được dùng các phép toán số học và logic.

d. Câu lệnh WITH: Khi cần truy nhập đến nhiều thành phần của một biến kiểu Record, ta có thể dùng câu lệnh WITH để chương trình được gọn hơn. Dạng câu lệnh WITH:

WITH <biến kiểu Record> do <Câu lệnh>

Câu lệnh này có thể lồng nhau để truy nhập vào các trường ở sâu trong Record.

e. Record có cấu trúc thay đổi

Ví dụ 2-9: Lập danh sách sinh viên, tuổi, điểm thi toán và in ra danh sách phải thi lại khi điểm thi < 5.

Program Danh_sach_sinh_vien_thi_lai;

```
uses crt;
type
        svien=record
              hoten: string;
              tuoi: string;
              diem: real;
              end;
var
        a: array[1..10] of svien;
        i,n: integer;
begin
              write('nhap hoten sinh vien:'); readln(a[i]. hoten);
              write('nhap tuoi:'); readln(a[i]. tuoi);
              write('nhap vao diem thi:'); readln(a[i]. diem);
              end;
         writeln('Danh sach sinh vien thi lai la:');
```

```
    for i:=1 to 3 do
        if a[i]. diem<5 then
    begin
        writeln('Ho va ten:', a[i]. hoten);
        writeln('Tuoi:', a[i]. tuoi);
        writeln('Diem:', a[i]. diem:4:2);
    end;
readln
end.
```

Lưu ý: Chương trình trên dùng cấu trúc a: array[1..10] có nghĩa là phạm vi ghi tối đa chỉ có 10 sinh viên . Khi dùng for i:=1 to 3 là ta dự định chỉ nhập số liệu có 3 sinh viên thôi.

Sau khi nhập số liệu, ta sẽ có kết quả như:

Danh sach sinh vien thi lai la:
Tran Quoc Duong
Tuoi: 19
Diem: 4.5

2-8 CHƯƠNG TRÌNH CON, HÀM VÀ THỦ TỤC

1) Khái niệm chung

Khi lập chương trình, ta thường gặp trường hợp có nhiều đoạn chương trình lặp lại nhiều lần, hoặc nội dung giống nhau chỉ thay đổi giá trị tính toán. Để tránh phải viết lại nhiều lần, ta chuyển chương trình đó thành chương trình con để khi cần thì gọi ra sử dụng.

Khi lập chương trình lớn, ta cũng chia thành nhiều chương trình con để đỡ rối, dễ sửa đổi bổ sung. Mỗi chương trình con sẽ xử lý một công việc nào đó. Chương trình con càng độc lập về dữ liệu, về các biến thì càng dễ sửa đổi chương trình.

Thí dụ khi tính lưu lượng mạng lưới thoát nước mưa, xem hình 4-4 và ví dụ 4-1, trình tự chung như sau:

- Chương trình con: tính lưu lượng Q_i tại giếng thăm G_i do lưu vực F_i gây ra

- Sau đó lặp lại tương tự: tính lưu lượng Q_2 tại giếng thăm G_2 do lưu vực F_2 gây ra

- Cuối cùng đưa ra kết quả: lưu lượng Q tại giếng thăm G_2 do cả F_1+ F_2 gây ra là $Q= Q_1+ Q_2$.

Sau đây giới thiệu lý thuyết chung để thực hiện bài toán trên.

2) Chương trình con: thủ tục

Một thủ tục bắt đầu bằng từ khoá Procedure và được tổ chức như sau:

```
Procedure<tên thủ tục>(tham số 1: kiểu, tham số 2: kiểu…);
    Var
            <khai báo các biến cục bộ>
    Begin
            <các lệnh>
    End;
            (*chương trình chính*)
    Begin
            <lời gọi thủ tục>(danh sách các tham số thực sự)
            <các lệnh>
    End.
```

Lưu ý: Sau từ khoá End cuối thủ tục là dấu chấm phẩy (;) chứ không phải dấu chấm (.) như cuối chương trình chính.

3) Chương trình con: Hàm

Một hàm bắt đầu bằng từ khoá Funtion và được tổ chức như sau:

```
Funtion<tên hàm>(tham số 1: kiểu, tham số 2: kiểu…): Kiểu hàm
    Var
            <khai báo các kiểu biến cục bộ>
    Begin
            <các lệnh>
            tên hàm:=<giá trị>;
    End;
            (*chương trình chính*)
```

```
Begin
        Lời gọi hàm;
        <tên hàm>(danh sách các tham số thực sự)
End.
```

4) Sự khác nhau giữa hàm và thủ tục

Sự khác nhau cơ bản giữa hàm và thủ tục là:

- *Hàm* có một giá trị thông qua tên hàm và hàm có thể tham gia vào các biểu thức tính toán.

- *Thủ tục* không cho một giá trị nào cả. Và đương nhiên, tên thủ tục không thể tham gia vào các biểu thức tính toán, so sánh.

Trong nhiều trường hợp, hàm tham gia như một chương trình độc lập. Trong thân hàm, bao giờ cũng có lệnh gán giá trị cho tên hàm.

Các tham số được ghi sau tên hàm và thủ tục là các tham số hình thức.

- Trong *thủ tục*, các tham số hình thức có 2 loại:

+ Các tham số khai báo sau từ khoá *Var* là các tham số biến.

+ Các tham số khai báo không có chữ *Var* là tham số giá trị.

- Trong *hàm* chỉ có tham số có giá trị.

Tham số thực sự là tham số dùng trong lời gọi hàm hay thủ tục. Trong thủ tục, các tham số giá trị thường là các biến để chứa dữ liệu đưa vào thủ tục, các tham số biến là các biến mà kết quả tính toán của thủ tục sẽ chứa vào đó. Khi ra khỏi thủ tục, ta vẫn có thể sử dụng chúng.

5) Biến toàn cục, biến cục bộ và truyền dữ liệu

a. Biến toàn cục

Biến toàn cục là biến khai báo ở đầu chương trình chính. Biến này tồn tại trong suốt thời gian tồn tại của chương trình. Các biến này có thể thay đổi giá trị ở chương trình chính cũng như trong các chương trình con.

b. Biến cục bộ

Biến cục bộ là biến được khai báo ở đầu chương trình con. Sau khi ra khỏi chương trình con thì biến này bị xoá. Biến cục bộ có giá trị trong chương

trình con và trong các chương trình con khác gọi từ chương trình con chứa biến này.

Nếu ta khai báo biến cục bộ trùng với biến toàn cục của chương trình chính thì máy vẫn không thông báo sai sót. Trong trường hợp này, máy vẫn chuẩn bị một ô nhớ khác để lưu trữ 2 biến. Khi ra khỏi chương trình con thì biến cục bộ được giải phóng.

Danh sách các tham số thực sự trong lời gọi phải tương ứng với các tham số hình thức trong khai báo chương trình con và chúng phải tương ứng về kiểu.

c. Truyền dữ liệu

Khi gặp một lời gọi tới chương trình con, máy sẽ thực hiện các bước sau:

- Cung cấp bộ nhớ cho các biến cục bộ

- Truyền các giá trị của các tham số trong lệnh gọi chương trình con (nếu có) các tham số tương ứng được khai báo trong lời gọi chương trình con.

- Thực hiện các lệnh trong chương trình con. Trong khi thực hiện các lệnh của chương trình con, các biến cục bộ có thể bị thay đổi nhưng không làm ảnh hưởng đến chương trình chính. Tham số trong chương trình con có 2 loại: Tham số giá trị và tham số biến. Các tham số trong lời gọi chương trình con và các tham số được khai báo trong chương trình con phải tương ứng với nhau.

Ví dụ 2-10. Sử dụng thủ tục.

```
Var
        s1,s2,tong:Interger;
Procedure Tinh;
    Begin
        tong:=s1+s2;
    End;
        (*Chương trình chính*)
    Begin
        Write('s1=');readln(s1);
        Write('s2=');readln(s2);
```

```
    Tinh;
        Wreadln(tong);
        Readln;
End.
```

Trong chương trình trên, ta đã không truyền tham số nào vào chương trình con mà sử dụng luôn các biến toàn cục là *s1, s2, tong* để tính toán các chương trình con. Vì biến toàn cục có ý nghĩa trong tất cả các chương trình con nên điều này không ảnh hưởng gì đến nội dung chương trình.

Ví dụ 2-11: Truyền tham số cho thủ tục. Nhập vào 4 số, cứ sau khi nhập xong hai số thì thông báo tổng của hai số này.

```
Var
        s1,s2,tong:Interger;
Procedure Tinh(a,b);
    Begin
        Tong:=s1+s2;
    End;
        (*Chương trình chính*)
    Begin
        Write('s1=');readln(s1);
        Write('s2=');readln(s2);
        Tinh(s1,s2);
        Writeln(tong);
        Readln;
        Write('s1=');readln(s1);
        Write('s2=');readln(s2);
        Tinh(s1,s2);
        Writeln(tong);
        Readln;
    End.
```

Khi ta thực hiện chương trình, nhập giá trị vào hai biến *s1, s2* và lời gọi thủ tục *Tinh(s1,s2)* sẽ truyền hai giá trị của biến *s1, s2* vào hai biến tương ứng trong thủ tục là *a* và *b*. Biến *tong* sẽ cộng hai giá trị trên lại. Trong các

biến trên, các biến sau là biến toàn cục: *s1, s2, tong* và các biến *a, b* là biến cục bộ. Các biến này chỉ tồn tại trong thu tuc *Tinh*. Nếu ta không khai báo biến *Tong* ở trong chương trình chính mà ở thủ tục *Tinh* như sau:

```
Procedure Tinh(a,b:Interger);
Var
      Tong:Interger
```

thì câu lệnh:

```
      Writeln(tong);
```

ở trong câu lệnh sẽ bị thông báo lỗi vì biến *Tong* ở trong thủ tục *Tinh* đã được giải phóng khi ra khỏi thủ tục này.

Ví dụ 2-12: Sử dụng hàm.

```
Var
      a,b:Interger;
Funtion Tinh:interger;
      Begin
      Tinh:=a+b;
      End;
      (*-----------*)
      Begin
            a:=1;b:=2;
            writeln(tinh);
            readln;
      End.
```

Kết quả nhận được là:

3

Trong chương trình trên, tên hàm đã chứa một giá trị là tổng của hai số a và b do lệnh:

```
      Tinh:=a+b;
```

quy định. Khi khai báo một hàm, ta phải khai báo kiểu hàm và kiểu dữ liệu này sẽ được truyền về chương trình chính cũng như dùng trong các phép toán trong hàm.

d. Truyền dữ liệu cho các tham số giá trị

Chúng ta xét chương trình sau đây:

```
Var
        tg:interger
Begin
        tg:=a;
        a:=b;
        b:=tg;
End;
        (*Kết thúc chương trình con, bắt đầu chương trình chính*)
Begin
        a:=11;b:=22;
        Đổi_chỗ(a,b);
        writeln('a=',a,'b=',b);
        readln;
End.
```

Mục đích của chương trình này là thay đổi giá trị của hai biến a và b. Và như ta đã thấy trong chương trình con đã có các lệnh để đổi giá trị của hai biến thông qua biến tg. Khi ta thực hiện chương trình thì kết quả nhận được như sau:

a=11 b=22

Như vậy đã không thực hiện được yêu cầu đề ra vì các biến này vẫn giữ nguyên giá trị ban đầu.

Cách khai báo trong chương trình con như trên gọi là khai báo theo tham số giá trị. Các biến khai báo theo dạng này có thể bị thay đổi trong chương trình con nhưng hoàn toàn không ảnh hưởng đến giá trị của biến đó (nếu có) ở chương trình chính.

Ta sẽ thực hiện được việc thay đổi giá trị nếu khai báo như sau:

```
Procedure doi_cho(a,b:interger);
```

Cách khai báo tham số như trên gọi là tham số biến sẽ được xét ở mục dưới đây.

e. Truyền dữ liệu cho các tham số biến

Ta hãy xét ví dụ sau đây:

```
Var
        a,b:Interger;
Procedure Tinh(Var x,y:interger);
Begin
        x:=x*10;
End;
        (*-------------------*)
Begin
        a:=11;b:=22;
        Tinh(a,b);
        Writeln('a=', a,' b=', b)
        Readln;
End.
```

Kết quả nhận được khi thực hiện chương trình như sau:

a=110 b=22

Như ta đã thấy ở trên, biến a tăng lên 10 lần do lệnh

x=x*10

Biến x và y trong chương trình con được khai báo sau từ khoá *Var* và được gọi là hàm số biến. Việc thay đổi giá trị của nó sẽ làm cho biến tương ứng với nó (biến a) thay đổi theo.

Ví dụ 2-13. Sử dụng chương trình con, lập chương trình giải phương trình bậc hai:

$$a.x^2 + b.x + c = 0$$

Nội dung chương trình:

```
Program Giai_phuong_trinh_bac_hai ;
Var
        aa, bb, cc, y1, y2: Real ;
        kq1: Interger;
Procedure ptbh(a,b,c: Real; var x1, x2: Real; var kq:interger);
```

```
Var
        Delta, căn: real;
Begin
    Delta:=b*b – 4*a*c;
    If delta=0 then
        Begin
        x1:=-b/(2*a); kq:=1;
        End;
    If delta>0 then
        Begin
        kq:=2; căn:=sqrt(delta);
        x1:=(-b+căn)/(2*a); x2:=(-b-căn)/(2*a);
        End;
End;
        (***********)
Begin
        Write('vào số a=');readln(aa);
        Write('vào số b=');readln(bb);
        Write('vào số c=');readln(cc);
    ptbh(aa, bb, cc, y1, y2, kq1);
    If kq1=0 then
        Writeln('Phương trình vô nghiệm');
    If kq1=1 then
            Begin
                Writeln('Phương trình có nghiệm kép');
                Writeln('x1=x2='y1:10:2);
            End;
    If kq1=2 then
            Begin
                Writeln('Phương trình có hai nghiệm phân biệt');
                Writeln('x1=',y1:10:2); writeln('x2=',y2:10:2);
            End; Readln;
End.
```

6) Phương pháp đệ quy

a. Định nghĩa

Đệ quy là kỹ thuật lập trình mà từ một điểm trong thân chương trình con lại gọi chính nó. Phương pháp đệ quy là một phương pháp ngắn gọn, nhưng do có nhiều lần gọi chương trình nên tốn bộ nhớ và thời gian thực hiện.

b. Ví dụ 2-14.

Lập chương trình tính giai thừa của một số n. Khi dùng lệnh *For* ta sẽ lập như sau:

```
        Program Tính_giai_thừa;
Var
        n:interger;
Funtion Giai_thua: Longint;
Var
        i:interger;
        Gt:Longint;
Begin
        Gt:=1;
        If n>0 then
     For i:=1 to n do Gt:=Gt*i;
        Giai_thua:=Gt;
End;
        (*---------------*)
Begin
        Writeln('Nhập số n=');Readln(n);
        Writeln(Giai_thua);
        Readln;
End.
```

Khi sử dụng đệ quy, ta có thể lập chương trình như sau:

```
Var
        n:interger;
```

```
Funtion Giai_thua(n:interger):Longint;
Var
        Gt:Longint;
    Begin
        If n=0 then giai_thua:=1 else
            Giai_thua:=n*Giai_thua(n=1);
    End;
        (*----------------*)
Begin
        Write('Nhập n=');Readln(n);
        Writeln(Giai_thua(n));readln;
End.
```

2-9. LỆNH GỌI THỦ TỤC, HÀM VÀ THOÁT RA KHỎI CHƯƠNG TRÌNH

1) Lệnh gọi thủ tục và hàm

Câu lệnh gọi thủ tục hoặc hàm là câu lệnh gọi thực hiện thủ tục hoặc hàm chuẩn, hay thủ tục hoặc hàm tự tạo trước đó. Câu lệnh này gồm có ***Tên thủ tục*** hoặc ***Tên hàm***, tiếp theo có thể *có hoặc không* danh sách các ***tham số*** đặt giữa hai dấu ***ngoặc đơn*** (). Danh sách các tham số là một dãy các ***Biến*** và ***Biểu thức*** được đặt cách nhau bởi ***dấu phẩy*** (,). Trong quá trình thực hiện chương trình nếu gặp câu lệnh gọi thủ tục hay hàm, điều khiển được chuyển đến thủ tục hay hàm có ***tên*** được chỉ định, đồng thời giá trị các tham số được truyền cho thủ tục hay hàm theo các cách: ***Truyền bằng trị, truyền bằng biến*** hoặc ***truyền kết hợp*** (vừa trị vừa biến). Sau khi thực hiện xong thủ tục hoặc hàm, chương trình thực hiện tiếp tục từ câu lệnh *đứng ngay sau* câu lệnh gọi thủ tục hoặc hàm.

Ví dụ 2-15. Câu lệnh gọi thủ tục và hàm

```
    Program Lệnh_gọi_thủ_tục_và_hàm;
Uses Crt;
Var
    n:Interger;
    FUNTION Giai thừa(x:Interger):LongInt;      {Hàm tự tạo}
```

```
Var
     i:Interger;
     gt:LongInt;
Begin
     gt:=1;
          For i:=1 To x Do
               Gt:=gt*i;
     Writeln('+Giai thừa của',x,'=',gt);
     End;
BEGIN
     ClrScr;                        {Gọi Thủ tục chuẩn để xoá màn hình}
     Writeln('LENH GOI THU TUC VA HAM'); {Lệnh gọi thủ tục chuẩn}
     Writeln('----------------------');    {Lệnh gọi thủ tục chuẩn}
     Write('Nhap so nguyen N=');           {Lệnh gọi thủ tục chuẩn}
     Readln(n);                            {Lệnh gọi thủ tục chuẩn}
     Writeln('+Binh phuong cua',n,'=',Sqr(n));
                                           {Lệnh gọi thủ tục và hàm chuẩn}
     Writeln('+Can bac hai cua',n,'=',Sqrt(n):0:2);
     Giaithua(n);                          {Lệnh gọi hàm tự tạo}
     Writeln;                              {Lệnh gọi thủ tục chuẩn}
     Writeln('Bam phim <Enter> de ket thuc'); {Lệnh gọi thủ tục chuẩn}
     Readln;                               {Lệnh gọi thủ tục chuẩn}
END.
```

2) Thủ tục thoát khỏi chương trình

Trong lập trình, nhất là các chương trình dài, hoặc ngay cả những chương trình con, đôi lúc cũng cần dừng chương trình tại một vị trí nào đó để kiểm tra các kết quả trung gian mà không phải chạy hết chương trình. Để thực hiện điều này Pascal cung cấp cho ta 2 thủ tục sau:

- ***Thủ tục Exit:*** Nếu thủ tục Exit đặt trong chương trình con thì thoát chương trình con giữa chừng để trở về chương trình chính. Nếu exit đặt trong chương trình chính thì sẽ chấm dứt chương trình chính, không thực hiện các lệnh còn lại sau thủ tục này.

- ***Thủ tục Halt:*** Dù cho thủ tục này đặt trong chương trình con hay chương trình chính, khi gặp thủ tục này sẽ chấm dứt thực hiện chương trình để trở về hệ điều hành và xuất hiện một thông báo mã thoát *(Exit Code). Halt* hoặc *Halt(0)* là thoát về hệ điều hành không có lỗi.

Chương trình dưới đây minh hoạ những điều trình bày ở trên. Ta hãy nạp vào máy và chạy thử.

Ví dụ 2-16. Sử dụng thủ tục *Exit* hoặc *Halt* để thoát khỏi chương trình hoặc chương trình con.

```
Program Thoat_chuong_trinh;
Var
        N:Interger;
        Ch:Char;
Begin
      Repeat
           Write('Bam mot phim, "t" de thoat');
           Readln(ch);
           If (ch='t') or (ch=,T,) then
                Exit
           Elsse
         Begin
                Write('-Nhap mot so nguyen n=');
                Readln(n);
                Writeln('+Binh phuong cua',n,'=',Sqr(n));
                Writeln('+Can bac hai cua',n,'=',Sqrt(n):0:2);
          End;
       Write('-Thuc hien'tiep khong (c/k)');
       Readln(ch)
           Until (ch='k') or (ch='K');
 End.
```

Khi chạy chương trình nếu ngay từ đầu ta gõ chữ 't' hoặc 'T' thì sẽ chấm dứt chương trình, bỏ qua các lệnh còn lại.

Chương 3

LẬP TRÌNH TURBO PASCAL THIẾT KẾ ỔN ĐỊNH NỀN ĐƯỜNG

3-1. KHÁI QUÁT CHUNG. THUẬT NGỮ CHÍNH

1) Khái quát chung

Để thiết kế, kiểm tra sự ổn định của nền đường, từ đơn giản đến phức tạp, thường có các nội dung sau:

- Kiểm tra ổn định mái ta luy
- Thiết kế tường chắn ở nơi đắp cao, đào sâu, ven hồ, sườn núi dốc
- Tính toán độ lún, tốc độ lún của nền thiên nhiên dưới nền đường đắp
- Kiểm tra sự ổn định của nền thiên nhiên dưới nền đường đắp
- Nếu thấy không ổn định, phải đề ra biện pháp gia cố nền (bằng cọc cát, bấc thấm, vải địa nhân tạo v.v...)

Về kiểm tra sự ổn định của nền thiên nhiên dưới nền đường đắp, là một bài toán hay gặp khi thiết kế đường trên vùng đất yếu (vùng đồng bằng Bắc, Trung, Nam Việt Nam). Có nhiều phương pháp như: kiểm tra theo cung trượt Bishop, kiểm tra theo đường đẳng trị ứng suất (Ko) để xác định vùng biến dạng dẻo...

Trong chương này, sẽ đề cập đến phương pháp kiểm tra, thiết kế ổn định nền thiên nhiên dưới nền đắp bằng vùng biến dạng dẻo, tính độ lún, tốc độ lún v.v...

Các vấn đề sẽ được trình bày theo trình tự: tóm tắt lý thuyết chung, phương pháp tính toán bằng tay, phương pháp lập trình tính toán theo Turbo Pascal.

Chú ý: Khi lập trình trong Turbo Pascal chỉ dùng được chữ Việt không dấu để diễn giải, nhưng để dễ hiểu, đôi khi tác giả vẫn dùng chữ Việt có dấu trong chương trình Pascal.

2) Tóm tắt lý thuyết cơ học đất

Để nắm vững bản chất các bài toán tính ổn định nền đường trong chương này, ta cần tìm đọc ở các tài liệu chuyên sâu về lý thuyết cơ học đất. Để hỗ trợ, sau đây chỉ nêu tóm tắt một số nội dung chính về cơ học đất.

Tuy nhiên, cũng cần nhấn mạnh thêm rằng: việc nắm vững lý thuyết cơ học đất sẽ giúp ta phân tích nhiều "trường phái", quan điểm tính toán khác nhau, qua đó cũng hiểu rõ các chương trình tính toán (phần mềm – Software) của máy tính đã được lập có ưu khuyết gì, và ta sẽ tự lập chương trình theo cách tính riêng mà ta cho là hợp lý...

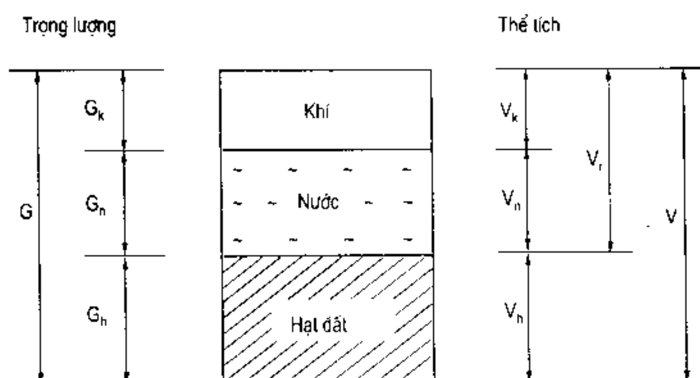
a. Các thành phần hợp thành đất và tác dụng lẫn nhau giữa chúng.

Đất gồm 3 thành phần chính như hình 3-1:

+ Thành phần hạt

+ Nước

+ Khí



Hình 3-1: *Các thành phần của đất*

Các chỉ tiêu cơ lý thông dung của đất được xác định bằng công thức định nghĩa, tính toán, liên hệ như trong bảng 3-1.

Bảng 3-1. Các chỉ tiêu cơ lý thông dụng của đất

Thứ tự	Chỉ tiêu	Định nghĩa công thức cơ bản	Công thức liên hệ – Ghi chú
1	Trọng lượng riêng của đất	$\gamma = \frac{G}{V}$	
2	Trọng lượng riêng ướt	$\gamma_w = \frac{G_h + G_n}{V}$	
3	Trọng lượng riêng no nước (Gn' là trọng lượng nước khi đầy lỗ rỗng)	$\gamma_{nn} = \frac{G_h + G'_n}{V}$	
4	Trọng lượng riêng đẩy nổi (γ_n trọng lượng riêng của nước, = 1)	$\gamma' = \frac{G_h + \gamma_n V_k}{V}$	$\gamma' = \frac{(\Delta - 1)\gamma_n}{1+e} = \gamma_{nn} - \gamma_n$
5	Trọng lượng riêng khô	$\gamma_k = \frac{G_h}{V}$	$\gamma_k = \frac{\gamma_w}{1 + 0{,}01W}$, còn gọi là γ_k là γ_c, W tính bằng %
6	Trọng lượng riêng của hạt đất	$\gamma_h = \frac{G_h}{V_h}$	
7	Tỉ trọng hạt đất (hoặc đặc hoàn toàn)	$\Delta = \frac{G_h}{\gamma_h V_h}$	$\Delta = \frac{\gamma_h}{\gamma_n} \rightarrow \gamma_n = 1 \rightarrow \Delta = \gamma_h$
8	Độ ẩm của đất	$W\% = \frac{G_n}{G_h}100$	Thường dùng như độ ẩm tự nhiên
9	Độ no nước của đất (số thập phân)	$G = \frac{V_n}{V_r}$	$G = \frac{0{,}01W\Delta}{e}$ $= \frac{\Delta\gamma.0{,}01W}{\Delta\gamma_n(1 + 0{,}01W) - \gamma}$
10	Độ rỗng	$n\% = \frac{V_r}{V}100$	$n\% = \frac{e}{1+e} \times 100$, n càng bé càng chặt
11	Hệ số rỗng	$e = \frac{V_r}{V_h}$	$e = \frac{\gamma_h}{\gamma_k} - 1$ $= \frac{\Delta\gamma_n(1 + 0{,}01n)}{\gamma}$

Bảng 3.1 (tiếp theo)

Thứ tự	Chỉ tiêu	Định nghĩa công thức cơ bản	Công thức liên hệ – Ghi chú
12	Độ ẩm bão hòa	$W_{bh} = \frac{e}{\Delta}$	$W_{bh} = \frac{\Delta - \gamma_k}{\Delta\gamma_k}$
13	Giới hạn sệt, hoặc nhão, chảy	W_l	Thí nghiệm bằng độ lún của chùy
14	Giới hạn dẻo	Wp	Thí nghiệm bằng lăn tay
15	Chỉ số dẻo	$I_p = W_L - W_p$	
16	Chỉ số nhão (Độ sệt)	$I_L = \frac{W - W_p}{I_p}$	γ_{kmax} còn gọi là dung trọng khô lớn nhất (MDD) thí nghiệm bằng cối proctor
17	Độ chặt của đất	$K_c = \frac{\gamma_k}{\gamma_{k\,max}}$	Còn gọi $K_c = \frac{\gamma_c}{\gamma_c\,max}$ hoặc ODD
18	Độ chặt của cát	$D = \frac{e_{max} - e}{e_{max} - e_{min}}$	$1{,}00 \geq D > 0{,}67$ là cát chặt $0{,}33 \geq D > 0$ là cát xốp
19	Gradien thủy lực	$J = \frac{\Delta h}{l}$	Δh - tổn thất cột nước l - chiều dài đường thấm
20	Định luật thấm Darcy	Q = kFJt	F - diện tích vuông góc qua đường thấm
21	Hệ số thấm	k = Q/FJE	t - thời gian thấm - Q = vFt đơn vị cm/sec, cm/năm, đất sét $k = 1 \times 10^{-7}$ cm/sec
22	Độ lún biến dạng tương đối	$\lambda_D = \frac{S}{h}$	S - độ lún toàn bộ mẫu thí nghiệm h - chiều cao mẫu thí nghiệm
23	Mô đun biến dạng	$E_o = \frac{\sigma_z}{\lambda_z}$	$E_o = \beta E_{dh} = \frac{1 + e_1}{a}\beta$ σ_z - áp lực thẳng đứng lên mẫu đất
24	Độ lún đàn hồi tương đối	$\lambda = \frac{S_{dh}}{h}$	S_{dh} - độ lún đàn hồi của mẫu đất

Bảng 3.1 (tiếp theo)

Thứ tự	Chỉ tiêu	Định nghĩa công thức cơ bản	Công thức liên hệ – Ghi chú
25	Mô đun đàn hồi	$E = \frac{\sigma_z}{\lambda_z^{dh}}$	Thông thường $E_{dh} = (2,5 - 3,0)E_o$
26	Hệ số nén hông	$\xi = \frac{\sigma_k}{\sigma_z}$	Xét ảnh hưởng thí nghiệm mẫu không cho nở hông
27	Hệ số nở hông	$\mu = \frac{\xi}{1+\xi}$	Xác định để tìm các trị số áp lực đất phù hợp với thực tế
28	Hệ số nén lún	$a = \frac{e_1 - e_2}{p_2 - p_1}$	Đơn vị cm^2/kG là quan hệ biến dạng và lực tương tự định luật Hook
29	Độ lún toàn phần S	$S = \frac{a}{1+e_1}\Delta P.h$	$S = \frac{Ph}{E_o}\beta$ dùng khi đất biến dạng ΔP (kG/cm^2) là áp lực $\beta = 1 - \frac{2\mu}{1-\mu}$; thường $\beta = 0,8$
30	Hệ số cố kết	$C_v = \frac{k(1+e_1)}{\gamma_n a}$	Đơn vị cm^2/sec, m^2/năm, m^2/yr e_1 - độ rỗng ban đầu
31	Độ cố kết toàn phần	$U = 1 - (1 - U_h)(1 - U_v)$	Đơn vị U% – U_h độ cố kết ngang U_v - độ cố kết đứng
32	Độ cố kết ở thời điểm "t" theo phương thẳng đứng	$U_v = \frac{S_t}{S}$	S_t - độ lún theo thời gian t. Yêu cầu $S_t = 0,9S$ thì cho phép thi công mặt đường. $S_t = U_v.S$
33	Thời gian	$T = \frac{C_v t}{h_p^2}$	h_p - bề dày tính toán lớp đất $h_p = H$ khi thoát nước 1 chiều $h_p = \frac{H}{2}$ khi thoát nước 2 chiều

Xét riêng về **tính chất cơ học của đất**, gồm các tính chất sau:

- *Tính thấm*: là đặc tính quan trọng của đất, nó ảnh hưởng đến khả năng chịu lực của đất và quá trình lún; nó còn tạo áp lực thuỷ động gây xói – xói mòn, trương nở ta luy. Tiêu biểu cho tính thấm là hệ số thấm K (cm/s, cm/năm)

- *Tính nén lún*: một đặc tính cơ học dễ nhận ra của đất là bị lún khi chịu tải trọng. Với đất dính, quá trình lún dưới tác dụng của tải trọng xảy ra khá lâu, vì nước bị ép đẩy ra rất chậm. Đó chính là hiện tượng lún cố kết, thường phải xét đến khi đắp nền đường qua vùng đất yếu và thiết kế cải tạo nền đất yếu.

Tiêu biểu cho tính nén lún là các chỉ tiêu hệ số nén lún a (cm^2/kG) (có nơi còn ký hiệu Cc), mô đun biến dạng E_o (kG/cm^2), mô đun đàn hồi E, cường độ chịu lực nén của nền đất σ (kG/cm^2).

- *Cường độ chống cắt của đất:* là chỉ tiêu quyết định giới hạn bền của đất để không bị nứt, trượt. Cường độ chống cắt của đất chủ yếu phụ thuộc vào góc nội ma sát φ (độ) và lực dính C (kG/cm^2).

Theo lý thuyết Coulomb (1773): cường độ chống cắt của đất theo công thức:

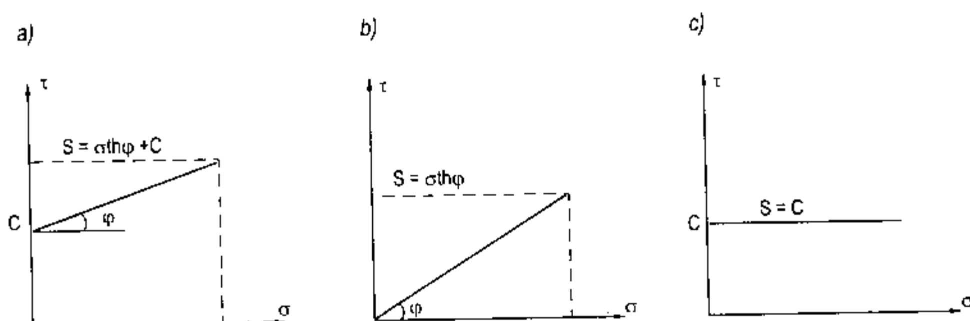
$$S = \sigma tg\varphi + C \qquad (3\text{-}1)$$

σ - ứng suất nén tác dụng lên mặt phẳng đang xét, (kG/cm^2);

φ - góc nội ma sát của đất, (độ);

C - lực dính, (kG/cm^2).

Hình 3-2: biểu diễn quan hệ công thức (3-1) với đất thông thường, cát và đất sét thuần tuý.



Hình 3-2: *Đồ thị quan hệ τ, C, φ*

a) Đất thông thường, b) Cát (C = 0), c) Đất sét thuần tuý (φ = 0)

Nền đất theo một mặt phẳng nhất định sẽ ổn định nếu ứng suất cắt τ trên mặt phẳng ấy nhỏ hơn S.

$$\tau < S \qquad (3\text{-}2)$$

Khi thiết kế ổn định nền đường, cải tạo nền đường (như bằng bấc thấm...) rất cần xác định được chỉ tiêu C, φ của nền đất. Hiện có nhiều phương pháp thí nghiệm C, φ và cũng có nhiều quan điểm khác nhau về cách dùng trị số C, φ khi tính toán.

Người ta đã rút ra kết luận rằng: dùng máy nén 3 trục để thí nghiệm chỉ số C, φ theo 3 trạng thái là phù hợp với các điều kiện thực tế nhất, bao gồm cả hiệu quả của áp lực nước lỗ rỗng. Tóm tắt như bảng 3-2.

Thí nghiệm nén 3 trục được tiến hành theo 2 giai đoạn:

1- Tạo áp lực hình cầu trong buồng nén σ_3, tương đương với một điểm trên vòng Morh

2- Tăng liên tục biến thiên của áp lực tới khi mẫu bị phá hoại theo một mặt phẳng nghiêng theo một góc α

Bảng 3-2. Phương pháp thí nghiệm C, φ và phạm vi áp dụng

Thứ tự	Giai đoạn		Thông số thu được	Tương ứng với tình trạng thực tế
	1-a	2-a		
1	U không cố kết	U không thoát nước	$\varphi_u = 0$ C_{uu}	Đắp nhanh trên nền chưa cố kết, thấm ít
2	C cố kết	U không thoát nước	$\varphi_{cu} = \varphi_d$ $C_{cu} = C_d$	Đối với đường đắp cao. Tính toán ổn định tức thời của ta luy
3	C cố kết	D thoát nước	φ_d C_d	Ổn định theo thời gian của ta luy. Đắp chậm theo giai đoạn trên nền thấm nước

***Ghi chú bảng 3-2**: U, C, D là chữ đầu tiếng Anh Unconsolidated, Undrained, Consolidated, Drained.*

3) Ký hiệu thuật ngữ chính dùng khi lập chương trình Pascal

Khi tính toán cơ học đất, nền móng ta đã quen thuộc với một số ký hiệu như γ, φ, ε, σ, τ...như bảng 3-1 trên. Nhưng khi lập trình Borland Turbo

Pascal (BP, TP) ta không thể dùng các ký hiệu đó được. Bảng 3-3 sau đây giới thiệu một số ký hiệu, thuật ngữ chính thường dùng. Một số khác sẽ giới thiệu trong từng công thức cụ thể sau.

Trong lập trình Pascal không phân biệt chữ hoa, chữ thường, tuy nhiên ta vẫn tự phân biệt chữ hoa, chữ thường để dễ nhìn.

Bảng 3-3. Ký hiệu, thuật ngữ chính

Ký hiệu	Đơn vị	Ý nghĩa
a, α	độ	Góc điểm tính toán nhìn đáy nền
r	radian	Góc điểm tính toán nhìn đáy nền (=pi*a/180)
C	kG/cm^2 T/m^2	Lực dính
C_c	cm^2/kG, m^2/T	Hệ số nén lún
C_v	cm^2/s, m^2/năm	Hệ số cố kết đứng
C_h	cm^2/s, m^2/năm	Hệ số cố kết ngang
e, ε	-	Hệ số rỗng
F, φ	độ	Góc nội ma sát của đất nền
f_1	radian	Góc nội ma sát của đất nền (=pi*F/180)
G, γ	g/cm^3, T/m^3	Trọng lượng đơn vị, dung trọng đất
G1, γ_{dn}	g/cm^3, T/m^3	Dung trọng đẩy nổi
Gt, Δ	-	Tỷ trọng hạt đất
B, σ	kG/cm^2, T/m^2	Ứng suất pháp tác động lên phân tố đất
K_v	cm/s	Hệ số thấm đứng
K_h	cm/s	Hệ số thấm ngang
pi, π	-	Số PI
S	cm, m	Độ lún cố kết
T_{max}, τ_{max}	kG/cm^2, T/m^2	Ứng suất tiếp lớn nhất của đất
t, tg	-	Hàm lượng giác tg (t = sinf1/ cosf1)
t	tháng, năm	Thời gian lún cố kết (công thức 3-)
U_h, U_v	%	Độ cố kết ngang, đứng

3-2. SỰ ỔN ĐỊNH CỦA NỀN THIÊN NHIÊN DƯỚI NỀN ĐƯỜNG ĐẮP

1) Lý thuyết xác định vùng biến dạng dẻo

a. Xác định sơ đồ tính toán tải trọng tác dụng do nền đắp và xe chạy

Tải trọng xe chạy thường tính 1,00 T/m^2 hoặc 0,10 kG/cm^2

Có 2 dạng sơ đồ thường dùng là:

+ Tải trọng hình băng phân bố đều

+ Tải trọng hình thang

Trong tính toán để đơn giản, có thể thay thế tải trọng tiết diện hình thang bằng hình chữ nhật tương đương, có cường độ tải trọng P_o phân bố đều trên đáy B = 2b + a.

b. Tính ứng suất nền và vùng biến dạng dẻo trường hợp tải trọng băng

Xét ứng suất một phân tố đất M bất kỳ nhìn đáy nền 2b một góc α. Ứng suất pháp thẳng đứng và nằm ngang tính theo công thức:

$$\sigma_z = \frac{P_o}{\pi}[\alpha - \sin\alpha.\cos(\alpha_1 - \alpha_2)] \quad (3\text{-}3)$$

$$\sigma_x = \frac{P_o}{\pi}[\alpha + \sin\alpha.\cos(\alpha_1 - \alpha_2)]$$

Ứng suất tiếp trên mặt nằm ngang và thẳng đứng:

$$\tau_{xz} = \tau_{zx} = \frac{P_o}{\pi}.\sin\alpha.\sin(\alpha_1 + \alpha_2) \quad (3\text{-}4)$$

Ứng suất tiếp lớn nhất

$$\tau_{max} = T_{max} = \frac{\sigma_1 - \sigma_3}{2} = \frac{P_o}{\pi}\sin\alpha = \frac{P_o}{\pi}\sin r \quad (3\text{-}5)$$

σ_1 và $\sigma_2 = \sigma_3$ (với bài toán phẳng) là các ứng suất chính

$$\sigma_1 = B_1 = \frac{P_o}{\pi}(\alpha + \sin\alpha) = \frac{P_o}{\pi}(r + \sin r) \quad (3\text{-}6)$$

$$\sigma_2 = \sigma_3 = B_3 = \frac{P_o}{\pi}(\alpha - \sin\alpha) = \frac{P_o}{\pi}(r - \sin r)$$

Để đánh giá độ ổn định nền thiên nhiên, ta thường dùng phương pháp đường đẳng trị K_o. K_o là hệ số ổn định chống trượt tại một điểm tính toán trong đất nền, trị số bằng:

$$K_o = 2\sqrt{A_o(A_o - tg\varphi)} = 2 \times Sqrt\left[A_o \times (A_o - t)\right] \quad (3\text{-}7)$$

Trong đó: $$A_o = \frac{\sigma_1 tg\varphi + C}{\sigma_1 - \sigma_3} = \frac{B_1 * t + C}{B_1 - B_3} \quad (3\text{-}8)$$

φ - góc nội ma sát của nền thiên nhiên

C - lực dính của nền thiên nhiên

Tại trục Z có $\sigma_1 = \sigma_z$

$\sigma_3 = \sigma_x$

Thực tế tính toán, ta dùng các trị số α khác nhau rồi tính σ_1, σ_3, A_o, K_o. Tại trị số α_o cho $K_o = 1$ ta xác định độ sâu H_o của vùng biến dạng dẻo (vùng có $K_o < 1$). Nếu $H_o < b$ là nền ổn định, $H_o > b$ là nền không ổn định.

c. Ứng suất nền khi lớp đất bị nén trên nền cứng

Trường hợp này ứng suất sẽ lớn hơn các trị số tính toán cho nửa không gian đồng nhất.

Nếu là tải trọng băng phân bố đều, theo tính toán của Egorov thể hiện trên bảng 3-4.

Trị số ứng suất tính theo công thức:

$$\sigma_z = K.P \quad (3\text{-}9)$$

K - hệ số tra bảng 3-4;

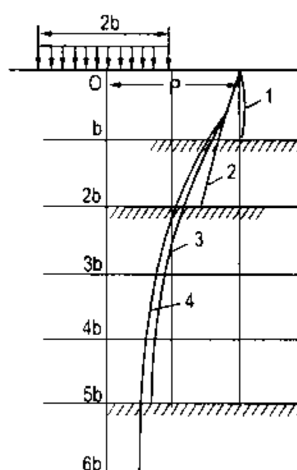
P - áp lực nền đắp lên mặt đất (kG/cm^2).

Bảng 3-4. Trị số K trong công thức (3-9)

Z/h	Khi lớp không biến dạng nằm ở độ sâu		
	h = b	h = 2b	h = 5b
1	*2*	*3*	*4*
1,0	1,000	1,00	1,00
0,8	1,009	0,99	0,82
0,6	1,020	0,92	0,57

1	2	3	4
0,4	1,024	0,84	0,44
0,2	1,023	0,78	0,37
0,0	1,022	0,76	0,36

Hình 3-3 thể hiện biểu đồ phân bố áp lực thẳng đứng khi có nền cứng (đường 1, 2, 3) và khi không có nền cứng (đường 4).



Hình 3-3: *Phân bố áp lực thẳng đứng trong nền khi có lớp không biến dạng nằm cao:*

1. *Khi độ sâu lớp không biến dạng là b*
2. *Khi độ sâu ấy là 2b;*
3. *Khi độ sâu là 5b*
4. *Đường phân bố ứng suất trong nửa không gian đồng nhất*

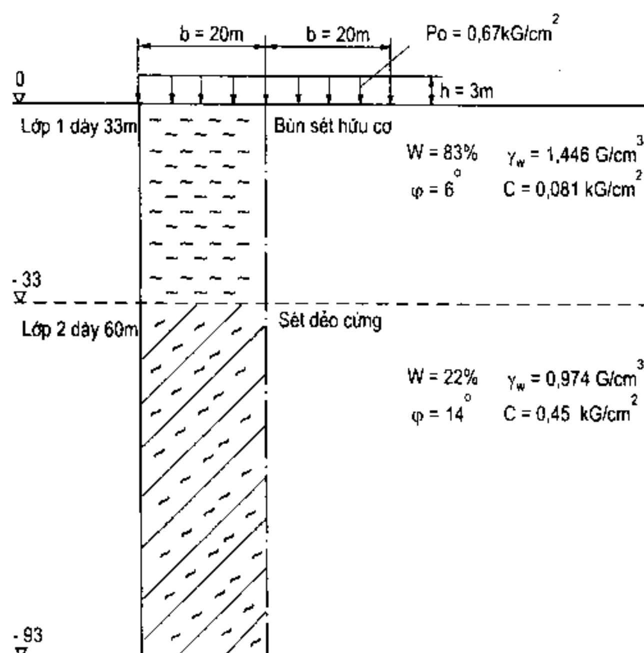
d. Tính ứng suất nền các trường hợp tải trọng khác

Ta còn có các trường hợp tính ứng suất cho các tải trọng khác như:

- Lực tập trung tác dụng thẳng đứng
- Tải trọng phân bố đều trên diện tích hình chữ nhật
- Tải trọng hình tam giác phân bố trên diện tích hình chữ nhật
- Tải trọng phân bố đều trên diện tích hình tròn và vành khuyên
- Tải trọng nằm ngang phân bố đều v.v...

Cách tính toán ứng suất được lập thành bảng, ta có thể tìm đọc trong một số sách về cơ học đất, thiết kế nền móng công trình, tính độ lún v.v... Tuy nhiên, do máy tính điện tử hiện nay đã khá phổ cập, ta có thể dựa vào công thức, lập mẫu bảng tính trên máy tính (như bảng tính trên Excel), sẽ tính nhanh ra các kết quả với nhiều thay đổi cần thiết của thông số đưa vào.

Ví dụ 3-1: Tính vùng biến dạng dẻo của nền thiên nhiên dưới nền đắp cao 3m bằng cát đen đầm chặt có $\gamma = 1.90$ T/m^3. Kích thước, địa tầng như bảng 3-5, hình 3-4.



Hình 3-4: *Nền đắp theo ví dụ 3-1.*

Bảng 3-5

Lớp	Tên đất	Bề dày m	Độ ẩm tự nhiên, W %	Dung trọng, γ_w, T/m^3	φ^o	Lực dính C, T/ m^2	Hệ số nén lún a, m^2/T	Hệ số thấm C_v, m^2/năm
1	Bùn sét hữu cơ	33	83	1,446	6	0,81	0,127	0,19
2	Sét dẻo cứng	60	22	1,974	14	4,50		

1. Xác định tải trọng P_o

- Tải trọng do cát đắp cao 3 m:

$$q_1 = 1,90 \times 3,00 = 5,70 \text{ T/m}^2 = 0,57 \text{ kG/cm}^2$$

- Tải trọng do xe chạy khi thi công $q_2 = 1,00$ T/ m^2 = 0,10 kG/cm^2

$$P_o = q_1 + q_2 = 5,7 + 1.0 = 6,7 \text{ T/ m}^2 = 0,67 \text{ kG/ cm}^2$$

2. Xác định toạ độ các điểm và tính toán

Vẽ nền đắp và địa tầng theo cùng tỷ lệ (xem hình 3-5), bước đầu "*cảm thấy*": với $\alpha = 50^o$ ở độ sâu khoảng 40 m là vùng nguy hiểm cần quan tâm. Vậy ta có thể lần lượt tính K_o với $\alpha = 50^o, 60^o, 70^o, 80^o, 90^o, 110^o, 140^o$ v.v... Tuy vậy, thực tế phải tính "*mò dần*" như thể hiện trên bảng 3-6 mới ra kết quả cuối cùng: vùng biến dạng dẻo có K < 1 nằm giữa 2 cung tròn $\alpha_2 = 25^o$ và $\alpha_{12} = 141^o$.

Riêng $\alpha_2 = 25^o$, là giới hạn dưới vùng biến dạng dẻo, lại nằm qua 2 lớp địa tầng khác nhau nên phải tính và vẽ vòng tròn đẳng trị K_o, σ_1, σ_3 (=σ_2), τ_{max} cho 2 lớp 1 và 2 khác nhau.

Áp dụng các công thức (3-5), (3-6), (3-7), (3-8) kết quả tính toán điển hình bằng tay như sau:

$$\alpha_5 = 50^o, \text{ đổi } \alpha_5 \text{ ra radian} = \frac{\pi \times 50}{180} = 0{,}8722, \sin 50^o = 0.7660$$

Lớp 1 có: $\varphi = 6^o$, $tg\varphi = 0{,}1051$, $C = 0{,}081$ kG/cm^2

$$\sigma_1 = B_1 = \frac{P_o}{\pi}(\alpha_1 + \sin\alpha_1) = \frac{0{,}67}{3{,}14}(0.8722 + 0.7660) = 0.3495$$

$$\sigma_3 = \sigma_2 = B_3 = \frac{P_o}{\pi}(\alpha_1 - \sin\alpha_1) = \frac{0{,}67}{3{,}14}(0{,}8722 - 0{,}7660) = 0{,}0227$$

$$\tau_{max} = T_{max} = \frac{P_o}{\pi}\sin\alpha_1 = \frac{0{,}67}{3{,}14} \times 0{,}7660 = 0{,}1634$$

$$A_o = \frac{\sigma_1 tg\varphi + C}{\sigma_1 - \sigma_3} = \frac{0{,}3495 \times 0{,}1051 + 0{,}081}{0{,}3495 - 0{,}0227} = 0{,}3602$$

$$K_o = 2\sqrt{A_o(A_o - tg\varphi)} = 2\sqrt{0{,}3602(0{,}3602 - 0{,}1051)} = 0{,}606$$

Với $\alpha_2 = 25^o$ qua 2 lớp:

Lớp 1 có: $\varphi = 6^o$, $tg\,\varphi = 0{,}1051$, $C = 0{,}081$ kG/cm^2.

Tính tương tự như trên được $K_o = 1.002$

Lớp 2 có: $\varphi = 14^o$, $tg\varphi = 0{,}2493$, $C = 0{,}450$ kG/cm^2.

Tính tương tự như trên được $K_o = 5{,}224$.

Như vậy có nghĩa là: thực tế vùng biến dạng dẻo chỉ nằm đến độ sâu hết địa tầng 1 (lớp bùn sét hữu cơ có chiều dày 33m) và trong cung tròn nhìn

đáy nền đắp = 2b một góc $\alpha = 25^o$, (cách tính độ sâu cung tròn sẽ theo công thức 3-10).

Trên hình 3-5 vòng tròn $\alpha_2 = 25^o$ có cùng trị số σ_1, σ_3, τ_{max} nhưng cung tròn nằm ở lớp địa tầng 2 sẽ có $K_o = 5,244$; hai nhánh 2 bên có $K_o = 1,002$. Tổng hợp kết quả tính toán như bảng 3-6.

Để xác định độ sâu H của đường đẳng trị K_o, ta áp dụng công thức sau:

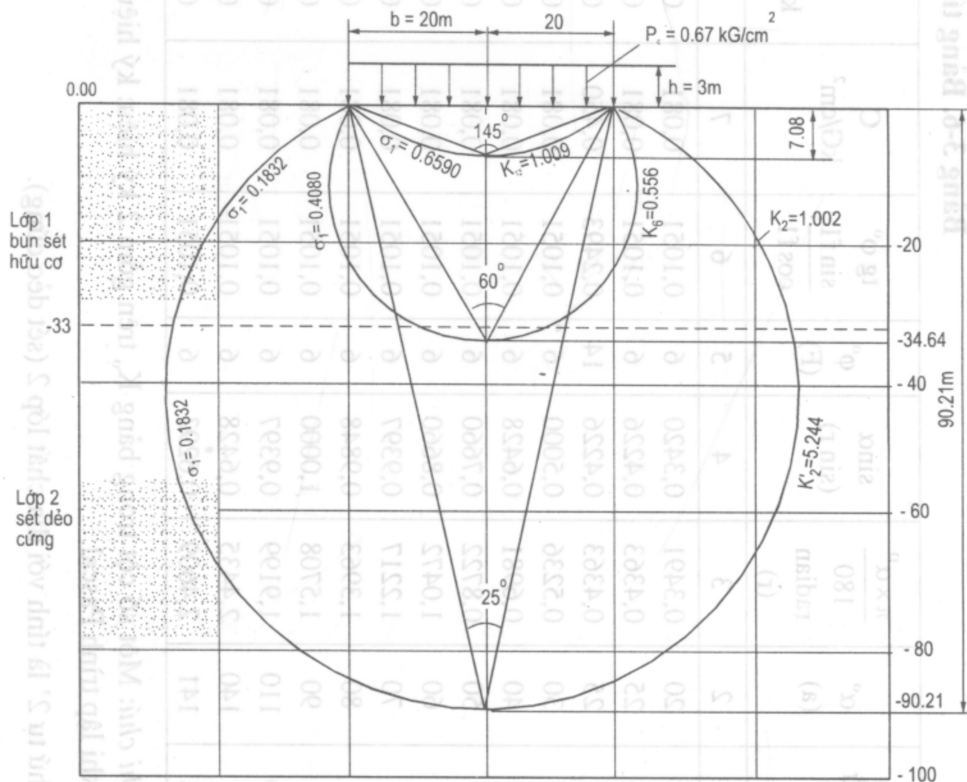
- Tính theo góc α^o $$H = b.tg\frac{180^o - a}{2} \quad (3\text{-}10)$$

- Tính theo góc α radian $$H = b.\frac{\sin(\pi - r)}{\cos(\pi - r)} \quad (3\text{-}11)$$

Kết quả tính với giới hạn trên và giới hạn dưới vùng biến dạng dẻo, tức khi $a = 25^o$ và 141^o là:

$$H_2 = 20 \times tg(77^o30') = 90,21 \text{ m}$$

$$H_{12} = 20 \times tg(19^o30') = 7,08 \text{ m}$$



Hình 3-5: Đồ thị đường đẳng trị σ_1, σ_2, τ_{max}, K_o theo ví dụ 3-1

Bảng 3-6. Bảng tính K_o

Thứ tự	α^o (a)	$\frac{\pi \times \alpha^o}{180}$ radian (r)	sinα (sin r)	φ^o (F)	tg φ^o $\frac{\sin f1}{\cos f1}$	C kG/cm²	σ_1 (B1) kG/cm²	σ_3 (B3) kG/cm²	τ_{max} (T_{max}) kG/cm²	A_o	K_o
1	2	3	4	5	6	7	8	9	10	11	12
1	20	0,3491	0,3420	6	0,1051	0,081	0,1474	0,0015	0,0729	0,661	1,213
2	25	0,4363	0,4226	6	0,1051	0,081	0,1832	0,0029	0,0901	0,556	1,002
2'	25	0,4363	0,4226	14	0,2493	0,450	0,1832	0,0029	0,0901	2,750	5,224
3	30	0,5236	0,5000	6	0,1051	0,081	0,2183	0,0050	0,1066	0,487	0,863
4	40	0,6981	0,6428	6	0,1051	0,081	0,2860	0,0118	0,1371	0,405	0,697
5	50	0,8722	0,7660	6	0,1051	0,081	0,3495	0,0227	0,1634	0,360	0,606
6	60	1,0472	0,8660	6	0,1051	0,081	0,4080	0,0386	0,1847	0,335	0,556
7	70	1,2217	0,9397	6	0,1051	0,081	0,4610	0,0601	0,2004	0,323	0,531
8	80	1,3963	0,9848	6	0,1051	0,081	0,5078	0,0878	0,2100	0,320	0,524
9	90	1,5708	1,0000	6	0,1051	0,081	0,5483	0,1217	0,2133	0,325	0,535
10	110	1,9199	0,9397	6	0,1051	0,081	0,6099	0,2090	0,2004	0,362	0,610
11	140	2,4435	0,6428	6	0,1051	0,081	0,6582	0,3840	0,1371	0,548	0,985
12	141	2,4609	0,6293	6	0,1051	0,081	0,6590	0,3906	0,1342	0,560	1,009

Ghi chú: Một số cột trong bảng K_o trên gồm 2 ký hiệu: ký hiệu trong ngoặc (như r, sin r, f1, B1, B3...) là để dùng cho khi lập trình Pascal.

Thứ tự 2' là tính với địa chất lớp 2 (sét dẻo cứng).

2) Lập chương trình Pascal NĐ 3-1 xác định vùng biến dạng dẻo

Trên cơ sở các công thức tính toán và ví dụ 3-1 trên, ta có thể lập chương trình Pascal tính vùng biến dạng dẻo K_o như sau:

Program Vùng_biến_dạng_dẻo_Ko_và_độ_sâu_H;

```
Uses crt;
Var
        r,F,f1,C,b,B1,B3,Tmax,t,Ao,Ko,Po,H: real;
        a,n: integer ;
Begin
Clrscr ;
        Write('nhập vào 1/2 bề rộng nền b(m):'); readln(b);
        Write('nhập vào tải trọng Po(kG/cm2):'); readln(Po);
        Write('nhập vào góc nội ma sát F(degree):'); readln(F);
        Write('nhập vào lực dính C(kG/cm2):'); readln(C);
        Write('nhập vào số góc n(degree):'); readln(n);
For a:=1 to n do
    Begin
        r:=pi*a/180;
        B1:=Po*(r+sin(r))/pi ;
        B3:=Po*(r-sin(r))/pi ;
        Tmax:=Po*sin(r)/pi ;
        f1:=pi*F/180;
        t:=sin(f1)/cos(f1);
        Ao:=(B1*t+C)/(B1-B3);
        Ko:=2*sqrt(Ao*(Ao-t));
        H:=b*sin((pi-r)/2)/cos((pi-r)/2);
        Writeln('Trị số B1:',B1:6:4);
        Writeln('Trị số B3:',B3:6:4);
        Writeln('Trị số Tmax:',Tmax:6:4,' kG/cm2');
        Writeln('Trị số Ao:',Ao:6:3);
        Writeln('Trị số Ko:',Ko:6:3);
```

```
        Writeln('Trị số góc a la:',a:4,' degree');
        Writeln('Độ sâu vùng biến dạng dẻo là H(m):',H:6:2,' m');
    End;
  Readln
  End.
```

Một số điểm cần lưu ý:

- Chương trình NĐ1 được lập với nhiều lệnh ghi kết quả tính toán (Writeln(...)) cho B1, B3, Tmax, Ao..., để dễ kiểm tra kết quả bảng tính 3-6. Thực tế khi đã quen, ta chỉ cần ghi và in 3 kết quả Ko, a, H là đủ, như vậy chương trình sẽ gọn hơn.

- Khi chạy chương trình NĐ 3-1, vì màn hình chỉ đủ chỗ hiện ra một số kết quả ứng với một số góc độ, nên ta cũng phải cho dần trị số n để tìm ra góc a có K_0 xấp xỉ 1. Tuy nhiên, việc làm này khá nhanh, vì máy tính với góc chênh lệch từng 1^o nên có thể tìm ra ngay 2 trị số góc $a_2 = 25^o$ và $a_{12} = 141^o$.

- Hơn nữa, dùng chương trình này "*chạy thử*" với nhiều tình huống khác nhau về địa chất (như với nhiều trị số C, φ của nhiều phương pháp thí nghiệm khác nhau), bề dày địa tầng, cao độ nền đắp..., ta có thể nghiên cứu, nhận biết quy luật biến thiên vùng biến dạng dẻo với rất ít công sức tính toán.

- Chương trình NĐ 3-1 là chương trình cơ bản đầu tiên, tìm ra vùng biến dạng dẻo (còn gọi là vùng hoạt động H_o), làm cơ sở cho các bước tính toán tiếp theo như tính độ lún, tốc độ lún, cải tạo nền đất yếu bằng cọc cát, bấc thấm v.v...

- Vì các chương trình con loại này "*để dùng riêng*", thỉnh thoảng lại dùng đến, nên khi lập, phần nhập số liệu nên ghi rõ cả tên thông số, đơn vị... để dễ gợi nhớ và tránh nhầm lẫn đơn vị đo lường.

3) Lý thuyết tính độ lún nền thiên nhiên dưới nền đường đắp.

Nền thiên nhiên dưới nền đường đắp chịu tác động của 2 tải trọng (theo trục Z) là:

$P_1 = \sigma_s = \gamma.h$ do trọng lượng bản thân nền thiên nhiên, (T/m^2)

$P_n = \sigma_z$ do tải trọng nền đắp và hoạt tải, (T/m^2)

Tải trọng tổng cộng là $P_2 = P_1 + P_n$

Ở đây, $P_n = P_o$ như ở ví dụ 3-1.

Độ lún được tính theo công thức:

$$S = \frac{(e_1 - e_2)h}{1 + e_1} \tag{3-12}$$

Theo định luật nén lún, hệ số nén lún a xác định bằng thí nghiệm và tính theo công thức:

$$C_c = tg\alpha = \frac{e_1 - e_2}{P_2 - P_1} = \frac{e_1 - e_2}{P_n} \tag{3-13}$$

Rút ra:

$$S = \frac{C_c.P_n}{1 + e_1}h \tag{3-14}$$

e_1 - hệ số rỗng tự nhiên của nền, chịu áp lực P_1;

e_2 - hệ số rỗng khi chịu áp lực P_2;

h - chiều dày mỗi lớp đất tính toán (m), quy định không lớn quá 0,4 bề rộng đáy nền đường đắp phía trên.

Phạm vi tính lún (khu vực hoạt động của nền thiên nhiên) quy định ở độ sâu H_o có:

$$\sigma_z = (0.1 - 0.2)\ \sigma_s \tag{3-15}$$

Trên trục z, trị số $\sigma_z = P_n$ tính theo công thức (3-6)

$$\sigma_z = \sigma_1 = B_1 = \frac{P}{\pi}(\ \alpha + \sin\alpha\ ) = \frac{P_n}{\pi}\ (r + \sin r)$$

Do σ_s, σ_z tính trung bình cho mỗi lớp nên góc α = a sẽ là góc ở giữa mỗi lớp (trên trục Z), nhìn vào đáy nền đắp như hình 3-5.

Độ sâu vùng hoạt động H_o còn một quan điểm khác là lấy bằng độ sâu vùng biến dạng dẻo, xác định được trên hình 3-5, (theo ví dụ 3-1, khi toàn bộ địa tầng đều là bùn sét hữu cơ thì $H_o = H_2 = 90.21$ m). Tuy nhiên, ta có thể kết hợp cả 2 cách để chọn trị số H_o được hợp lý.

Ví dụ 3-2: Tính độ lún của nền thiên nhiên trên hình 3-4.

Chia các lớp h = 6.5m < 0.4 (2b) = 0.4 × 40 m = 16m.

Xác định góc α = a ở giữa mỗi lớp trên trục Z. Độ sâu đến giữa mỗi lớp là Hn.

Vì nền thiên nhiên nằm trong mức nước ngầm, tính trọng lượng riêng đẩy nổi theo công thức:

$$\gamma_{dn} = \gamma' = \frac{(\Delta - 1).\gamma_n}{1 + e_1} = \frac{(G_t - 1)}{1 + e_1} = G_1 \quad (3\text{-}16)$$

$\Delta = G_t$ - tỷ trọng hạt đất = 2.575;

$\gamma_n = G_n = 1$ - trọng lượng riêng của nước;

e_1 - hệ số rỗng tự nhiên của nền .

Lập bảng tính áp lực nền $P_1 = \sum \gamma'.h$ và $P_n = \sigma_z = \sigma_1 = B_1$ và độ lún S như sau:

Bảng 3-7. Bảng tính toán ví dụ 3-2

Lớp	Độ sâu tính toán Hn (m)	α^o a	γ' G_1 (T/m^3)	P_1 (T/m^2)	P_n (T/m^2)	C_c (m^2/T)	e_1	Độ lún, S (m)
1	3,25	161,54	0,46	1,49	6,69	0,1196	2,443	1,510
2	9,75	128,02	0,46	4,46	6,45	0,1196	2,443	1,455
3	16,25	101,81	0,46	7,43	5,88	0,1196	2,443	1,327
4	22,75	82,64	0,46	10,41	5,19	0,1196	2,443	1,172
5	29,25	68,73	0,46	13,38	4,55	0,1196	2,443	1,026

Cộng độ lún: 6,490 m

4) Lập chương trình Pascal NĐ 3-2: Tính độ lún

Trên cơ sở các công thức tính toán và ví dụ 3-2 trên, ta có thể lập chương trình Pascal tính độ lún S như sau:

```
            Program Tính_độ_lún_cố_kết_S ;
uses crt;
var
        b, Hs, Hn, h, e1, Gt, Gn, G1, Cc, P1, Po, Pn, r, a, S, t: real ;
        n, i: integer ;
```

```
begin
clrscr;
        write('nhập vào bề rộng b(m):'); readln(b) ;
        write('nhập vào tỷ trọng hạt Gt:'); readln(Gt) ;
        write('nhập vào độ rỗng tự nhiên e1:'); readln(e1) ;
        write('nhập vào hệ số nén lún Cc(m2/T):'); readln(Cc) ;
        write('nhập vào số lớp n:'); readln(n) ;
 for i:=1 to n do
    begin
        Po: = 6.7 ;
        h: = 6.5 ;
        Hs: = i*h ;
        Hn: = Hs – h/2 ;
        t: = b/Hn ;
        r: = 2*arctan(t) ;
        a: = 180*r / pi ;
        Gn: = 1 ;
        G1: = (Gt – 1) * Gn / (1+e1) ;
        P1: = G1*Hn ;
        Pn: = (Po / pi) * (r + sin(r)) ;
        S: = Cc*Pn*h / (1+e1) ;
      writeln('Dung trọng đẩy nổi của nền đất G1:', G1: 6: 2,' T/m3') ;
      writeln('Ứng suất do trọng lượng bản thân nền P1:', P1: 6: 2,' T/m2') ;
      writeln('Ứng suất do nền đắp và hoạt tải Pn:', Pn: 6: 2,' T/m2') ;
      writeln('Độ lún cố kết lớp n là S:', S: 6: 3,' m') ;
      writeln('Góc a:', a: 8: 2,' degree') ;
    end;
  readln
  end.
```

Ghi chú cho NĐ 3-2:

- Trị số Gn = 1 là trọng lượng riêng của nước.

- Các số Po = 6,7 T/m^2, chia lớp tính độ lún h = 6,5m đã xác định nên ta ghi luôn vào chương trình sau "begin", khi Po và h thay đổi, ta hoàn toàn có thể ghi theo trị số mới, như vậy sẽ đỡ phải lập thêm dòng write('nhập vào...:');

- Tham số tính toán t = b/ Hn = tg (a/2) là số trung gian để dùng hàm artan(r) tìm ra trị số r radian cần dùng trong công thức tính Pn.

- Chú ý phân biệt Hs là độ sâu đến đáy lớp n, còn Hn là độ sâu đến tim lớp n.

- Đặc điểm sử dụng NĐ 3-2 là khi địa tầng thay đổi ở lớp nào, (thí dụ tương ứng với n = 4), ta nhập các trị số Gt, e1, Cc của lớp 4 và sẽ lấy riêng kết quả độ lún của lớp 4, loại bỏ kết quả lớp 1, 2, 3.

Chạy chương trình NĐ 3-2:

Nhập các số liệu b = 20 m

Gt = 2,575

e1 = 2,443

Cc = 0.,1196 m2/T

n = 5

Sẽ cho ta kết quả đúng như bảng 3-6.

5) Lý thuyết tính tốc độ lún

Tổng độ lún cố kết của nền thiên nhiên dưới nền đường đắp (không kể độ lún trong bản thân nền đắp) là:

$$\sum S = S_{c1} + S_{c2} \tag{3-17}$$

S_{c1} là độ lún cố kết giai đoạn 1 (sơ cấp). Đây là giai đoạn nền bị nén chặt để đạt độ ẩm không đổi W = const, là độ lún quan trọng nhất cần tính tốc độ lún để định thời hạn thi công mặt đường, tức là xác định thời gian ổn định $T_{ôd}$. Thường $T_{ôd}$ ứng với lúc độ lún đạt 80 - 90% S_{c1}.

S_{c2} là độ lún cố kết giai đoạn 2 (thứ cấp) do biến dạng từ biến gây ra.

Trong xây dựng đường, ta thường quan tâm tính tốc độ lún S_{c1}.

$$S_{c1} = S_1 + S_2 \tag{3-18}$$

S_1 là lún tức thời, thường bằng (0,1 – 0,3) S_2

S_2 là lún do trọng lượng nền đắp phía trên và hoạt tải (xe chạy) mà ta đã tính theo công thức (3-14):

$$S = \frac{C_c.P_n.h}{1 + e_1}$$

Sau thời gian ổn định $T_{ôđ}$, độ lún St được xác định theo công thức:

$$St = U_v . S \qquad (3\text{-}19)$$

U_v là độ cố kết theo chiều đứng, dùng khi không có gia cố nền đất bằng cọc cát hay bấc thấm, cho ở bảng 3-8.

Bảng 3-8. Hệ số cố kết U_v theo các sơ đồ phân bố ứng suất

Nhân tố thời gian				
t	U_1	U_2	U_3	U_4
0,004	0,080	0,008		0,135
0,008	0,104	0,016		0,186
0,012	0,125	0,024		0,223
0,020	0,160	0,040		0,279
0,028	0,189	0,056		0,322
0,036	0,214	0,072		0,358
0,048	0,247	0,096		0,388
0,060	0,276	0,120		0,433
0,072	0,303	0,144		0,462
0,100	0,357	0,197		0,516
0,125	0,399	0,244		0,554
0,167	0,461	0,318	$U_3 = U_1 - \frac{l_1 - l_2}{l_1 + l_2}(U_1 - U_2)$	0,605
0,20	0,504	0,370		0,638
0,25	0,562	0,443		0,682
0,30	0,631	0,508		0,719
0,35	0,658	0,565		0,752
0,40	0,698	0,615		0,780
0,50	0,764	0,700		0,829
0,60	0,816	0,765		0,868
0,80	0,887	0,857		0,918
1,00	0,931	0,913		0,950
2,00	0,994	0,993		0,996
	1,000	1,000		1,000

Khi có cọc cát hay bấc thấm, nước thoát theo phương ngang là chính

$$St = U_h . S \tag{3-20}$$

Độ cố kết toàn bộ U tính theo công thức:

$$U = 1 - (1 - U_v)(1 - U_h) \tag{3-21}$$

U_h - độ cố kết theo chiều ngang, xác định theo đồ thị hình 3-6, xác định theo tỷ số:

$$n = \frac{l}{d} \tag{3-22}$$

l – khoảng cách cọc cát, (m)

d - đường kính cọc cát, (m)

và nhân tố thời gian

$$T_h = \frac{K_h(1 + e_{bq})t}{a.\gamma_n.l^2} = \frac{C_h t}{l^2} \tag{3-23}$$

C_h - hệ số cố kết theo phương ngang

Bảng 3-8 cho các trị số Uv cho những trường hợp điển hình về sự phân bố áp lực theo chiều sâu của lớp đất bị nén. Trị số U phụ thuộc vào nhân tố thời gian T.

$$T = \frac{C_v.t}{h_p^2} \tag{3-24}$$

Trong đó:

C_v - hệ số cố kết theo hướng thẳng đứng

$$C_v = \frac{K_v(1 + e_{bq})}{a.\gamma_n} \tag{3-25}$$

K_h, K_v là hệ số thấm theo hướng ngang và đứng;

e_{bq} - hệ số rỗng trung bình trong phạm vi áp lực từ P_1 đến P_2, (P_1 - áp lực do trọng lượng bản thân nền, P_2 - áp lực tổng cộng do bản thân nền thiên nhiên và tải trọng nền đắp, xe chạy);

a – hệ số nén lún (cm^2/kG hoặc m^2/T);

γ_n – trọng lượng riêng của nước (= 0,001 kG/cm^3 hoặc 1 t/m^3);

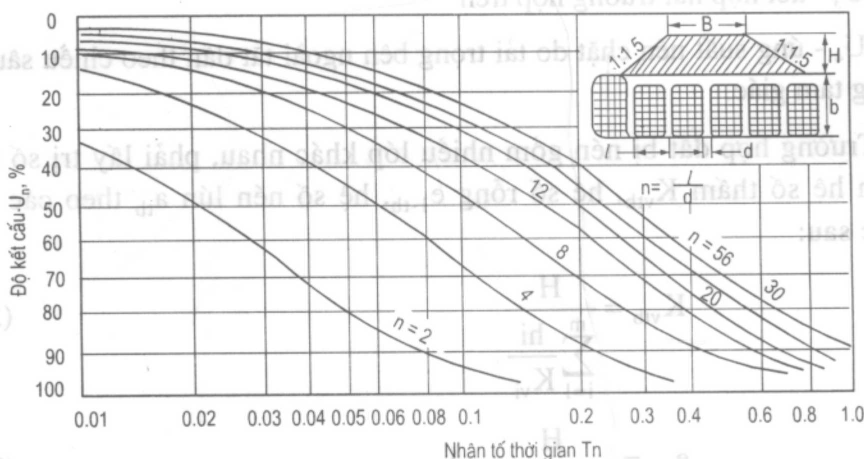
h_p – bề dày tính toán lớp đất chịu nén.

Khi thoát nước 1 chiều $h_p = H$

Khi thoát nước 2 chiều $h_p = H/2$

H là bề dày lớp nền thiên nhiên chịu nén, còn gọi là vùng hoạt động.

t là thời gian lún, có thể xác định qua thí nghiệm trong phòng với mẫu đất có chiều cao h_1.



Hình 3-6: *Đồ thị xác định độ cố kết ngang do tác dụng của giếng cát*

Mẫu đất này chịu áp lực bằng áp lực nền đắp, qua theo dõi lún theo thời gian của mẫu sẽ vẽ được đường cong nén lún theo thời gian. Từ đó rút ra quan hệ:

$$\frac{t_1}{t_2} = \left(\frac{h_1}{H}\right)^2 \tag{3-26}$$

t_1 – thời gian mẫu đất đạt được độ cố kết đã cho

t_2 – thời gian cần thiết để nền thiên nhiên đất yếu dày H đạt cùng độ cố kết ấy.

Cần chú ý:

Với nhiều loại đất số mũ trong công thức (3-26) không phải là 2 mà là n < 2

$$\frac{t_1}{t_2} = \left(\frac{h_1}{H}\right)^{n_o} \tag{3-27}$$

Trị số n_o xác định bằng thí nghiệm nhiều mẫu đất có chiều cao khác nhau.

- Trường hợp gia cố nền bằng giếng cát hoặc bấc thấm, trị số H sẽ thay bằng Dc là đường kính vùng ảnh hưởng của giếng cát hoặc bấc thấm.

- Trong bảng 3-8, cách vận dụng sơ đồ phân bố ứng suất như sau:

U_1 - đất nền bị nén chặt với ứng suất σ = const

U_2 - đất nền cố kết dưới tải trọng bản thân

U_3 - kết hợp hai trường hợp trên

U_4 - ứng suất nén chặt do tải trọng bên ngoài tắt dần theo chiều sâu nền, dạng tam giác.

Trường hợp đất bị nén gồm nhiều lớp khác nhau, phải lấy trị số trung bình hệ số thấm K_{vtb}, hệ số rỗng $e_{1\text{-}tb}$, hệ số nén lún a_{tb} theo các công thức sau:

$$K_{vtb} = \frac{H}{\sum_{i=1}^{m} \frac{hi}{K_{vi}}} \tag{3-28}$$

$$e_{bq} = \frac{H}{\sum_{i=1}^{m} \frac{hi}{1 + e_{qi}}} - 1 \tag{3-29}$$

$$a_{tb} = \frac{1 + e_{bq}}{H} - \sum_{i=1}^{m} \frac{a_i h_i}{1 + e_{1-tb}} \tag{3-30}$$

Trong đó:

H – bề dày toàn bộ các lớp đất bị nén

$$H = \sum_{i=1}^{m} h_i$$

h_i, K_{vi}, e_{bqi}, a_i là bề dày, hệ số thấm đứng, hệ số rỗng bình quân (giữa P_1 và P_2), hệ số nén của từng lớp riêng lẻ.

m – số lớp

Ví dụ 3-3: Tính và vẽ biểu đồ độ lún theo thời gian của nền đường đắp trên hình 3-4. Hệ số cố kết trung bình là:

Cv =3,71 × 10^{-4} cm^2/sec = 3,71 × 10^{-4} × π × 10^7 m^2/năm = 1,17 m^2/nằm

Kết quả tính toán qua ví dụ 3-2 cho S = 6,490m.

Do tải trọng Pn có dạng tam giác, áp dụng trị số U_4 ở Bảng 3-8.

Theo công thức (3-24) rút ra:

$$t = \frac{Th_p^2}{Cv} = \frac{T \times 16,5^2}{1,17} = 2,32,6923 \times T \text{ (năm)}$$

Vì nền thoát nước 2 chiều, dùng hp = H/2 = 33/2 = 16,5 m.

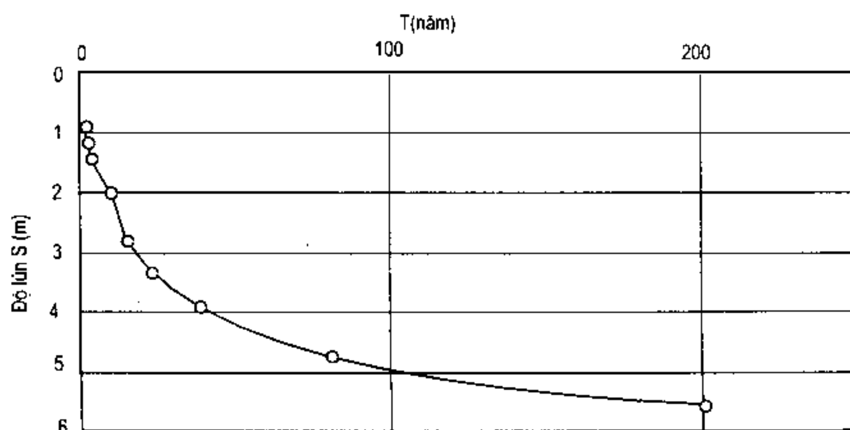
$H = H_a$ = 33 m là độ sâu vùng hoạt động, xét chỉ xảy ra ở lớp bùn sét hữu cơ, tìm được theo ví dụ 3-1.

Coi như nền đắp cao tức thời 3 m. Lập bảng tính như sau:

Bảng 3-9

U (Bảng 3-8)	0,135	0,186	0,223	0,322	0,433	0,516	0,605	0,752	0,868
T (Bảng 3-8)	0,004	0,008	0,012	0,028	0,060	0,100	0,167	0,350	0,600
t (năm)	0,9	1,8	2,8	9	14	23	39	81	202
St = US (m)	0,88	1,21	1,45	2,09	2,81	3,35	3,93	4,88	5,63

Theo kết quả bảng 3-9, dùng excel dễ dàng vẽ được biểu đồ tốc độ lún theo thời gian như hình 3-7.



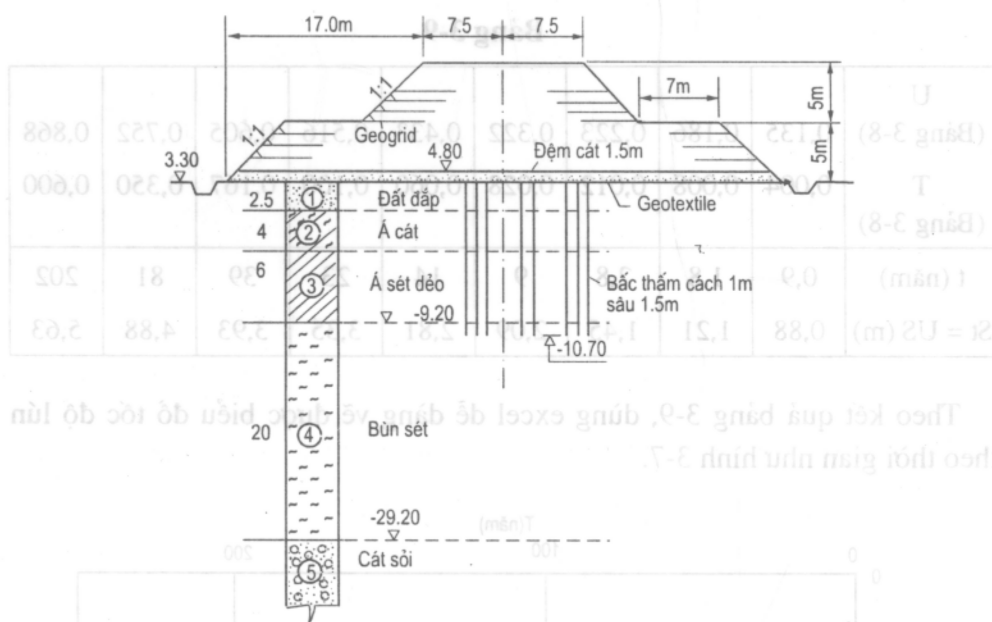
Hình 3-7: Biểu đồ tốc độ lún, ví dụ 3-3

6) Tính toán tốc độ lún khi cải tạo nền đất yếu bằng bấc thấm

Tính tốc độ lún theo ví dụ 3-3 là khi quá trình lún cố kết đạt được chủ yếu do thoát nước dọc từ nền thiên nhiên dưới nền đắp lên phía trên mặt đất tự nhiên (dưới chân nền đắp). Đây là quá trình thoát nước rất chậm, làm cho thời gian lún cố kết kéo dài hàng chục, hàng trăm năm.

Để rút ngắn thời gian cố kết, người ta thường dùng cọc cát hoặc bấc thấm để nước chủ yếu thoát ngang vào cọc cát, bấc thấm, chiều dài thấm trong đất ngắn, sau đó dễ dàng thoát dọc theo cọc cát, bấc thấm lên phía trên mặt đất tự nhiên.

Ví dụ 3-4: Một nền đắp trên nền đất yếu được gia cố bấc thấm (như hình 3-8), tóm tắt tính toán như sau:



***Hình 3-8:** Nền đắp gia cố bấc thấm*

1. Tính vùng hoạt động của nền đất yếu qua độ sâu H. Giả sử tính được H = 30m.

2. Tính đường kính tương đương của bấc thấm d_w:

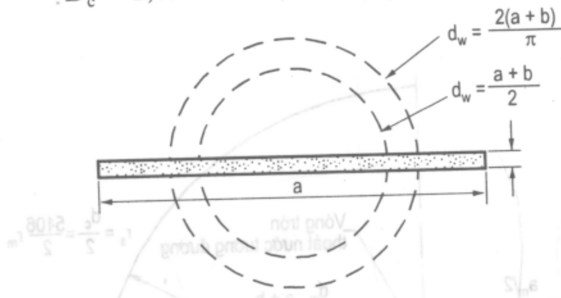
$$d_w = \frac{a + b}{2} \qquad (3\text{-}31)$$

Xem hình 3-9. Giả sử dùng loại bấc thấm có a = 100 mm, b = 4 mm

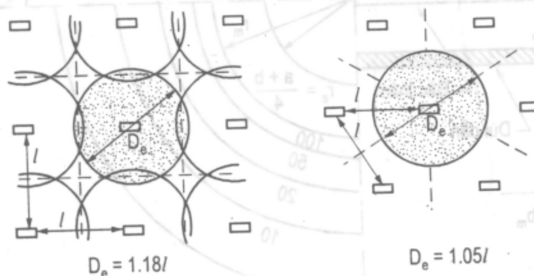
$$d_w = \frac{10 + 0,4}{2} = 5,2cm = 0,052m$$

3. Bấc thấm được bố trí theo hình tam giác đều, khoảng cách 2 bấc thấm là l = 1,00 m. Tính đường kính vùng ảnh hưởng của bấc thấm D_e:

$$D_e = 1,05 \times l = 1,05 \times 1,00 = 1,05m \quad (3\text{-}32)$$



Hình 3-9: *Đường kính tương đương của bấc thấm*



Hình 3-10: *Đường kính vùng ảnh hưởng của bấc thấm De theo mẫu đặt hình vuông và hình tam giác*

4. Tìm đường kính vùng xáo trộn d_s

Khi thi công bấc thấm, phải dùng một trục dùi dẫn rỗng giữa để đưa bấc thấm xuống độ sâu thiết kế như Hình 3-8. Do đó tạo nên một vùng xáo trộn có đường kính d_s và có ảnh hưởng đến quá trình thấm. Ảnh hưởng này gọi là ảnh hưởng nhiễm bẩn.

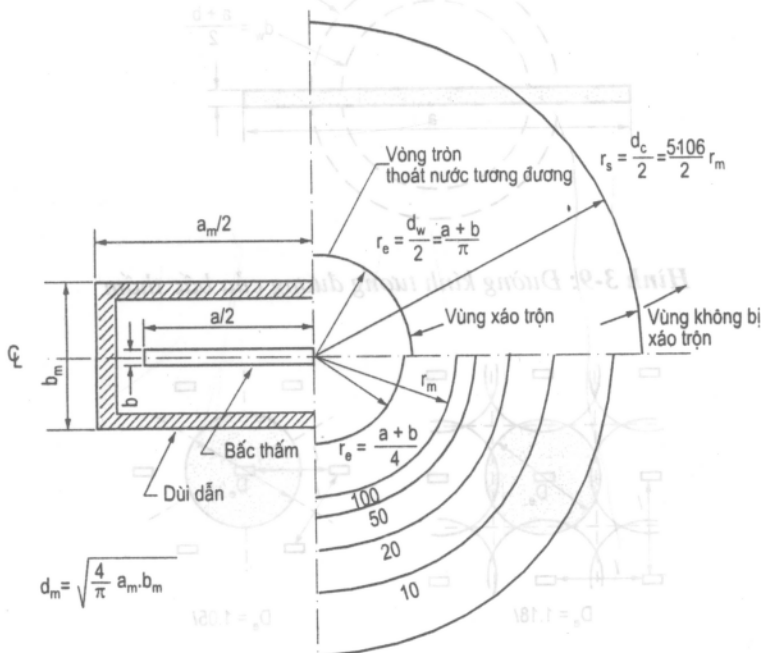
$$d_s = \frac{(5 \div 6)d_m}{2} \quad (3\text{-}33)$$

Hiện thường dùng theo khuyến nghị của Hansbo (1987):

$$d_s = 2d_m \quad (3\text{-}34)$$

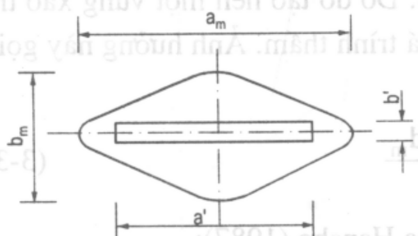
Trong đó d_m là đường kính vòng tròn tương đương của diện tích cắt ngang trục dùi dẫn. Khi dùi dẫn có tiết diện chữ nhật dài a_m, rộng b_m (Hình 3-11), d_m tính theo công thức:

$$d_m = \sqrt{\frac{4}{\pi} a_m . b_m} = \sqrt{\frac{4}{\pi} F_m} \qquad (3\text{-}35)$$



Hình 3-11:Xác định gần đúng vùng xáo trộn d_s quanh trục dùi dẫn

Thực tế Việt Nam còn dùng dùi dẫn tiết diện hình thoi (hình 3-12), $d_m = 9 \div 10$ cm.



a_m = 145 mm, b_m = 65 mm,

F_m = 61,6 cm^2, d_m = 9 cm.

a_m = 160 mm, b_m = 80 mm,

F_m = 80,0 cm^2, d_m = 10 cm.

***Hình 3-12:** Cắt ngang trục dùi dẫn hình thoi ở Việt Nam*

Thực nghiệm của Bergado (1991) cho thấy: dùng dùi dẫn loại nhỏ sẽ tăng nhanh tốc độ lún cố kết hơn loại lớn. Hệ số thấm K_s trong vùng xáo trộn bằng hệ số thấm đứng K_v.

$$K_s = K_v \tag{3-36}$$

5. Xác định tỷ số hệ số thấm ngang với hệ số thấm đứng K_h/ K_v

Trong đất gia cường bằng vải thô (như bấc thấm) rất dễ nhận ra rằng tỷ số Kh/ Kv có thể rất lớn, tới 10 lần. Tỷ số này có thể giảm bớt hoặc loại trừ do ảnh hưởng của nhiễm bẩn gây bởi vùng xáo trộn.

Do $K_s = K_v$ nên để xét ảnh hưởng nhiễm bẩn tới khả năng thấm nước, tức là thời gian cố kết, người ta dùng tỷ số K_h/ K_s.

Trong ví dụ này dùng $K = (K_h/ K_s) = 3$.

K_s càng nhỏ, thời gian cố kết càng lâu.

6. Hệ số cố kết ngang C_h

$$C_h = \frac{Kh}{Kv} C_v \tag{3-37}$$

Trong ví dụ này $C_h = 3C_v$

7. Tính thời gian cố kết

$$t = \frac{De^2}{8C_h}\left[\left(\ln\frac{De}{d_w} - 0,75\right) + \left(\frac{Kh}{Ks} - 1\right)\ln\frac{ds}{d_w}\right]\ln\left(\frac{1}{1 - U_h}\right) \tag{3-38}$$

Lập chương trình NĐ 3-3: Tính tốc độ lún (theo công thức 3-38)

Program Tinh_thoi_gian_lun_co_ket;

```
uses crt;
var
      t, De, Ch, dw, K, Kh, Ks, d, ds, S, St: real ;
      n, U: integer ;
begin
clrscr ;
      write('nhap vao so De(m):') ; readln (De);
      write('nhap vao so Ch(m2/năm):') ; readln (Ch);
      write('nhap vao so dw(m):') ; readln (dw);
```

```
write('nhap vao so K:') ; readln (K);
write('nhap vao so ds(m):') ; readln (ds);
write('nhap vao do lun toan bo S(m):') ; readln (S);
write('nhap vao tri so n:') ; readln (n);
   for U:=1 to n do
      begin
         d: =ds/ dw ;
         St: =U*S*0.1 ;
t: = ((sqr(De)/ (8*Ch))*((ln(De/ dw) – 0,75) + (K – 1)*ln(d))*ln(1/ (1 –
0.1*U)))*12;
         writeln('Thời gian lún cố kết là t (tháng):', t: 6: 2) ;
         writeln('Độ lún cố kết đạt được là St (m):', St: 6: 2) ;
      end ;
readln
end.
```

Chạy chương trình NĐ3-3:

Nhập các số liệu De = 1,05 m

Ch = 2,07 m^2/ năm

dw = 0,052 m

K = (Kh/ Ks) = 3

ds = 2dm = 0,20 m

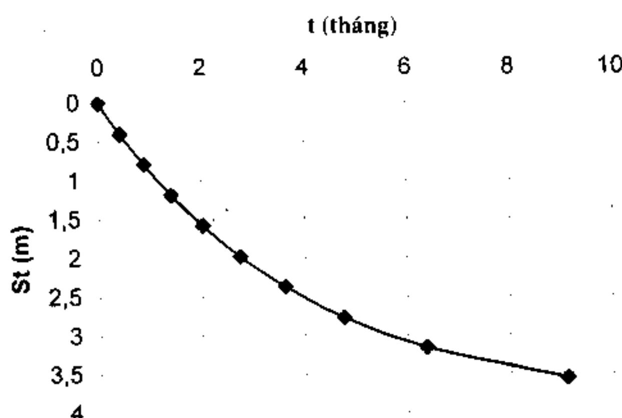
S = 3,952 m

n = 9

Được kết quả như bảng 3-10. Vẽ biểu đồ tốc độ lún khi có bấc thấm như hình 3-13.

Bảng 3-10. Kết quả tính toán độ lún theo thời gian

n	1	2	3	4	5	6	7	8	9
U (%)	10	20	30	40	50	60	70	80	90
t (tháng)	0.42	0,88	1,41	2,02	2,74	3,62	4,76	6,37	9,11
St (m)	0,40	0,79	1,19	1,58	1,98	2,37	2,77	3,16	3,56



Hình 3-13: *Biểu đồ tốc độ lún khi có bấc thấm, ví dụ 3-4*

Chú ý: Do Ch dùng đơn vị m^2/ năm nên t trong công thức (3-38) tính ra là năm. Để đổi ra đơn vị tháng, chương trình NĐ3-3 đã nhân với 12.

7) Một số nhận xét

Thực tế độ lún một con đường, hay cả một thành phố, qua hàng trăm năm còn chịu nhiều tác động khác như sự thay đổi phức tạp của mức nước ngầm, biến động kiến tạo vỏ trái đất v.v... Biểu đồ hình 3-7 chỉ có ý nghĩa về lý thuyết tính toán theo các kết quả thí nghiệm đất một khu vực riêng lẻ, dùng làm cơ sở để lý giải chứng minh các giải pháp cải tạo nền đất yếu mà chúng ta sẽ đề xuất.

Các kết quả tính toán trong ví dụ 3-2, 3-3 (như độ lún S = 6,49m, độ sâu vùng hoạt động khi toàn bộ là bùn sét hữu cơ H = 90,2m) là dựa trên số liệu địa chất thực tế của vùng lầy giữa sông lạch ngoại ô TP Hồ Chí Minh. Số liệu đó có vẻ như "vô lý", nhưng qua biểu đồ 3-7, bảng 3-9, trong 1,8 năm (=21,6 tháng) một nền đắp cao 3 m, rộng 40m bị lún 1,21 m là hoàn toàn có thể xảy ra.

Kinh nghiệm tính toán vùng biến dạng dẻo, độ sâu vùng hoạt động nền đất yếu thường trong khoảng H = 40 đến 45 m. Do đó, khi khảo sát địa chất phục vụ cho thiết kế cải tạo nền đất yếu, cũng phải khoan sâu 40 – 45 m mới đủ số liệu để tính toán.

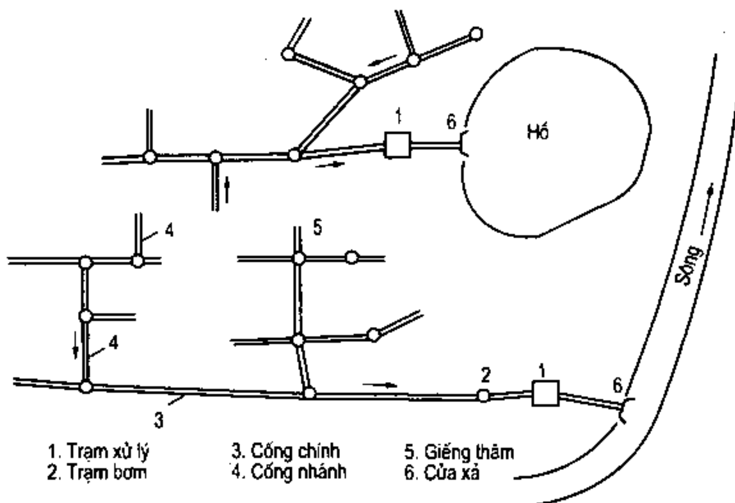
Chương 4

LẬP TRÌNH TURBO PASCAL THIẾT KẾ THOÁT NƯỚC

4-1. KÝ HIỆU, THUẬT NGỮ CHÍNH

1) Khái quát chung

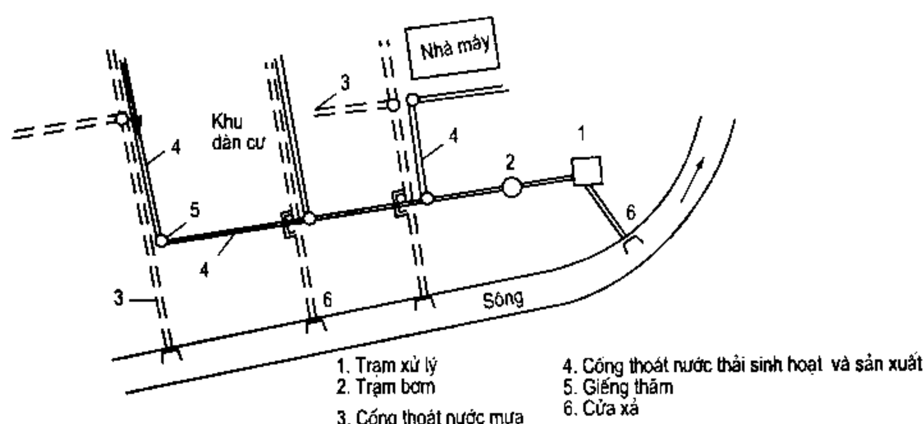
Ai đã từng tính toán thuỷ văn thuỷ lực cho cầu cống thoát nước đều thấy rằng: đó là một bài toán nhiều biến số phức tạp, thường phải tính "mò dần" và điều chỉnh nhiều lần, với khá nhiều loại công thức "số mũ thập phân", hoặc phải tra khá nhiều bảng tính, biểu đồ phức tạp (mà mỗi người tra ra một kết quả khác nhau), do đó khá "ngại" !



Hình 4-1: Hệ thống thoát nước chung

Với mạng lưới thoát nước đô thị, với hệ thống cống dọc thu nước từ giếng thu 2 bên hè, lại nối dọc nối ngang các phố trước khi đổ ra cửa xả (ở bờ sông), việc tính toán càng phức tạp hơn, (xem hình 2-1, 2-2). Lập chương trình phần

mềm tính toán cho mạng lưới thoát nước này là một bài toán rất phức tạp. Trong chương này, chỉ giới thiệu một số "chương trình con" để tham khảo, giúp cho mỗi kỹ sư cầu đường có thể tự lập chương trình riêng, theo phương pháp tính toán mà mình ưa thích, rồi "dùng riêng" trong máy tính cá nhân của mình. (Do vậy, phương pháp tính toán, công thức tác giả dùng cũng chỉ để tham khảo, vì mục đích chính là thể hiện ý tưởng lập trình).



Hình 4-2: Hệ thống thoát nước riêng

Chú ý: khi lập trình trong Turbo Pascal đều dùng chữ Việt không dấu để diễn giải, nhưng ở đây, đôi khi tác giả vẫn dùng chữ Việt có dấu để dễ hiểu.

2) Ký hiệu, thuật ngữ chính

Khi thiết kế, tính toán thuỷ văn thuỷ lực cho hệ thống thoát nước (cầu, cống...) ta đã quen thuộc với một số công thức và ký hiệu chữ La mã như ω, μ, ψ... Nhưng khi lập trình Turbo Pascal (TP), ta không thể dùng ký hiệu đó được. Sau đây giới thiệu một số ký hiệu, thuật ngữ chính thường dùng. Ngoài ra còn một số khác sẽ giới thiệu trong từng công thức cụ thể sau.

Bảng 4-1. Ký hiệu, thuật ngữ chính

Ký hiệu	Đơn vị	Ý nghĩa
a	-	Độ đầy nước chảy trong cống (=h/d)
d	cm, m	Đường kính cống
F_v	ha, km^2	Diện tích lưu vực
F_o, ω_o	m^2	Diện tích ướt

Bảng 4.1 (tiếp theo)

Ký hiệu	Đơn vị	Ý nghĩa
h	cm, m	Chiều sâu nước chảy
P	năm	Chu kỳ tràn cống
p_o, χ	m	Chu vi ướt
q	l/s.ha	Cường độ mưa
Q_m	l/s, m^3/s	Lưu lượng nước mưa
M_o, μ	-	Hệ số phân phối mưa rào
n	-	Hệ số nhám
R	cm, m	Bán kính thuỷ lực
V	m/s	Tốc độ nước
Y, ψ	-	Hệ số dòng chảy

Cần lưu ý rằng: vì lập trình Pascal không phân biệt chữ hoa, chữ thường nên để kết hợp với ký hiệu khi lập trình, các ký hiệu sẽ có nhiều ký tự phụ như F_v, F_o (phân biệt với f).

4-2 CHƯƠNG TRÌNH TN 4-1: TÍNH LƯU LƯỢNG THOÁT NƯỚC MƯA

1) Cường độ mưa tính toán. Thời gian mưa tính toán

Hiện nay vẫn tồn tại nhiều quan điểm, nhiều công thức khác nhau về tính cường độ mưa tính toán. Ở Việt Nam, qua số liệu nhiều năm, mỗi địa phương đã lập được quan hệ cường độ mưa và thời gian mưa với P = 0,5 – 20 năm, như hình 4-3 là thí dụ cho một địa phương.

Trên cơ sở xử lý số liệu thống kê cường độ mưa ở các tỉnh của Việt Nam trong 15 – 25 năm, Tiến sỹ Trần Hữu Uyển đã đưa ra công thức sau:

$$q = \frac{A_o(1 + C.\lg P)}{(t + b_o P^m)^n}, l/s.ha \qquad (4\text{-}1)$$

- Các thông số A_o, b_o, C, m, n cho ở bảng 4-2.

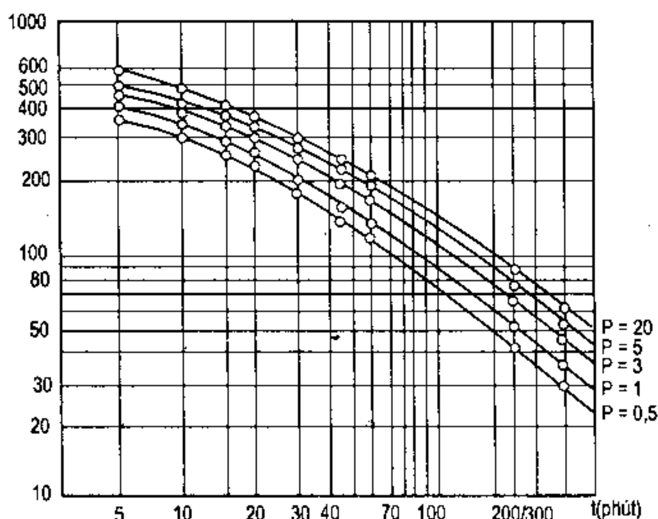
- P (năm) là chu kỳ tràn cống, là thời gian có một trận mưa vượt quá cường độ mưa tính toán.

+ Đối với khu dân cư, thành phố nhỏ P = 0,3 – 1 năm

+ Đối với thành phố lớn, khu công nghiệp lớn P = 1 - 3 năm

+ Đối với khu vực đặc biệt quan trọng P = 5 - 10 năm

- t (phút) thời gian mưa tính toán



***Hình 4-3:** Quan hệ giữa cường độ mưa và thời gian mưa*

Thời gian mưa tính toán là thời gian giọt nước mưa từ điểm xa nhất của lưu vực chảy đến tiết diện tính toán, còn gọi là thời gian cực hạn, (xem hình 4-4).

$$t = t_l + t_r + t_c \quad (4\text{-}2)$$

t_l - thời gian nước chảy từ điểm xa nhất đến rãnh thoát nước, còn gọi là thời gian tập trung nước bề mặt. Thường lấy $t_l = 5 - 10$ phút (min).

t_r - thời gian nước chảy trong rãnh đến giếng thu nước mưa gần nhất

$$t_r = 1.25\, l_r/V_r \quad (4\text{-}3)$$

l_r - chiều dài của rãnh (m);

V_r - tốc độ nước chảy trong rãnh (m/min);

1.25 - hệ số tính đến khả năng tăng tốc độ chảy trong quá trình mưa;

t_c - thời gian nước chảy trong cống từ giếng thu đến tiết diện tính toán

$$t_c = r.l_c/V_c \quad (4\text{-}4)$$

l_c - chiều dài đoạn cống tính toán (m);

V_c - tốc độ nước chảy trong cống (m/min).

r - hệ số phụ thuộc địa hình

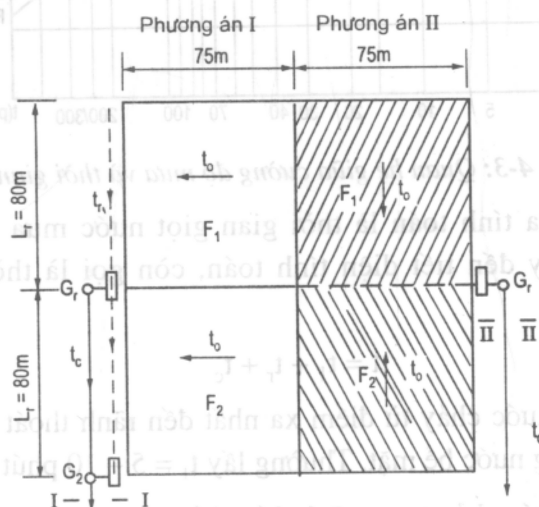
r = 2 khi địa hình bằng phẳng

r = 1.2 khi địa hình dốc hơn 3%

Vậy công thức 4-2 viết thành

$$t = t_1 + 1,25\frac{l_r}{V_r} + \sum\frac{L_c}{V_c} \quad (4\text{-}5)$$

Theo sơ đồ hình 4-4, để thoát nước mưa cùng diện tích $F_1 + F_2$, theo phương án 1 lưu lượng cực đại sẽ ở mặt cắt I-I qua 2 giếng thu, 2 giếng thăm. Thời gian mưa tính toán là $t = t_1 + t_r + t_c$. Theo phương án 2 lưu lượng cực đại sẽ ở II-II, qua 1 giếng thu, 1 giếng thăm. Thời gian mưa tính toán là $t = t_1 + t_r$.



Hình 4-4: Sơ đồ xác định thời gian mưa tính toán (cực hạn)

Bảng 4-2. Bảng thống kê các thông số khí hậu của công thức tính cường độ mưa của các tỉnh

Thứ tự	Tên tỉnh	A_o	b_o	C	m	n
1	Bắc Cạn	8150	27	0, 53	0, 16	0, 87
2	Bảo Lộc	11100	30	0, 58	0, 24	0, 95
3	Buôn Mê Thuột	4920	20	0,62	0,14	0,85
4	Cà Mau	9210	25	0,48	0,18	0,92

Bảng 4.2 (tiếp theo)

Thứ tự	Tên tỉnh	A_o	b_o	C	m	n
5	Đà Nẵng	2170	10	0, 52	0, 15	0, 65
6	Hà Giang	4640	22	0,42	0,20	0,79
7	Hà Nội	5890	20	0,65	0,13	0,84
8	Hòn Gai	3720	16	0,42	0,14	0,73
9	Hải Dương	4260	18	0,42	0,17	0,78
10	Hòa Bình	5500	19	0,45	0,18	0,82
11	Huế	1610	12	0,55	0,12	0,55
12	Lào Cai	6210	22	0,58	0,18	0,84
13	Lai Châu	4200	16	0,50	0,22	0,80
14	Móng Cái	4860	20	0,46	0,16	0,79
15	Nam Định	4320	19	0,55	0,18	0,79
16	Ninh Bình	4930	19	0,48	0,16	0,80
17	Nha Trang	1810	12	0,55	0,15	0,65
18	Hải Phòng	5950	21	0,55	0,15	0,82
19	Plâycu	7820	28	0,49	0,16	0,90
20	Phan Thiết	7070	25	0,55	0,16	0,92
21	Quảng Trị	2230	15	0,48	0,23	0,62
22	Quảng Ngãi	2590	16	0,58	0,12	0,67
23	Quy Nhơn	2610	14	0,55	0,18	0,68
24	Sơn La	4120	20	0,42	0,15	0,80
25	Sơn Tây	5210	19	0,62	0,17	0,82
26	Tuyên Quang	8670	30	0,55	0,12	0,87
27	Thái Nguyên	7710	28	0,52	0,20	0,85
28	Thái Bình	5220	19	0,45	0,16	0,81
29	Thanh Hóa	3640	19	0,53	0,15	0,72
30	Tuy Hòa	2820	15	0,48	0,18	0,72
31	Hồ Chí Minh	11650	32	0,58	0,18	0,95
32	Việt Trì	5830	18	0,55	0,12	0,85
33	Vinh	3430	20	0,55	0,16	0,69
34	Yên Bái	7500	29	0,54	0,24	0,85

2) Hệ số dòng chảy Y, (ψ)

Hệ số dòng chảy Y xét đến thực tế lượng mưa rơi xuống q_b chỉ chảy vào mạng lưới một phần q_c, còn lại thì thấm xuống đất hoặc bốc hơi.

$$Y = \psi = q_c/q_b \qquad (4\text{-}6)$$

Hệ số dòng chảy phụ thuộc vào tính chất bề mặt phủ, địa hình, độ dốc, thời gian mưa. Giáo sư P.C. Belốp đã đưa ra công thức thực nghiệm:

$$Y = Z.q^{0.2}.\,t^{0.1} \qquad (4\text{-}7)$$

Trong đó:

Z – hệ số thực nghiệm, đặc trưng cho tính chất của mặt phủ;

q – cường độ mưa (l/s.ha);

t – thời gian mưa (min).

Nếu diện tích bề mặt không thoát nước trên 30% thì có thể coi Y không đổi (nghĩa là không phụ thuộc vào q và t). Trị số Z và Y phụ thuộc vào tính chất bề mặt phủ cho trong bảng 4-3.

Bảng 4-3. Hệ số dòng chảy của các loại mặt phủ

Loại mặt phủ	Z	Y
f1- Mái nhà và mặt phủ bằng bê tông át phan	-	0.95
f2- Mặt phủ bằng đá dăm	0.224	0.60
f3- Đường lát đá cuội	0.145	0.45
f4- Mặt phủ bằng đá dăm không có vật liệu dính kết	0.125	0.40
f5- Đường sỏi trong vườn	0.090	0.30
f6- Mặt đất	0.064	0.20
f7-Mặt cỏ	0.038	0.10

3) Tính toán lưu lượng nước mưa Q_m

Theo phương pháp cường độ giới hạn của P.F. Gorbatrep: lưu lượng nước mưa ở tiết diện tính toán đạt giá trị cực đại khi thời gian mưa bằng thời gian nước chảy từ điểm xa nhất của lưu vực thoát nước tới tiết diện tính toán. Công thức tính toán là:

$$Q_m = M_o.Y.\,F_v.\,q\,, \qquad l/s \qquad (4\text{-}8)$$

$$Q_m = M_o.Y.F_v.A_o \frac{(1 + C.\lg P)}{(t + b_o P^m)^n} \quad (4\text{-}9)$$

F_v - diện tích lưu vực tính theo biểu đồ mặt bằng (ha)

M_o - hệ số phân bố mưa rào, đặc trưng cho sự phân bố mưa không đều trong lưu vực tính toán.

$$M_o = \frac{1}{1 + 0{,}001.F_v} \quad (4\text{-}10)$$

Khi lập trình theo Pascal, có đặc điểm là không có logarit thập phân và hàm số mũ thập phân, chỉ có logarit tự nhiên (ln) và hàm e^x (viết exp(x)). Do vậy công thức 4-9 cần biến đổi theo ln và e^x.

Biết rằng: $$\lg P = \frac{\ln P}{\ln 10} \quad (4\text{-}11)$$

$$P^m = \exp(m.\ln P) \quad (4\text{-}12)$$

Đặt: $t_2 = t + b_o.P^m$

$(t_2)^n = (t + b_o.P^m)^n$

Rút ra: $$q = A_o \frac{\left(1 + C.(\ln P / \ln 10)\right)}{\exp(n.\ln t_2)} \quad (4\text{-}13)$$

Ví dụ 4-1: Tính lưu lượng thoát nước mưa ở mặt cắt I-I theo hình 4-4. Địa điểm công trình là khu dân cư ở Hà nội. Diện tích $F_1 = F_2 = 0{,}60$ ha. Độ dốc mặt 0,006. Diện tích mặt phủ mái nhà 32%, mặt đường bê tông át phan 38%, mặt đường sỏi 8% và mặt lát cỏ 22%.

Nội dung tính toán bằng tay như sau:

- Xác định P: P = 1.

- Xác định q: áp dụng công thức (4-1). Theo bảng 4-2 với thành phố Hà nội có $A_o = 5890$, $b_o = 20$, $C = 0{,}65$, $m = 0{,}13$, $n = 0{,}84$.

$$q = 5890 \frac{(1 + 0{,}65 \lg 1)}{(t + 20.1^{0{,}13})^{0{,}84}} = \frac{5890}{(t + 20)^{0{,}84}}$$

- Xác định hệ số dòng chảy Y. Hệ số phân phối mưa rào M_o.

Khi có nhiều dạng mặt phủ, hệ số dòng chảy Y tính trung bình theo công thức sau:

$$Y = \frac{f_1.Y_1 + f_2Y_2 + ... + f_nY_n}{100} \quad (4\text{-}14)$$

$f_1, f_2, ... f_n$ là % diện tích mặt phủ

$Y_1, Y_2,... Y_n$ là trị số theo loại mặt phủ, cho ở bảng 4-3.

$$Y = \frac{32kt0,95 + 38 \times 0,95 + 8 \times 0,30 + 22 \times 0,10}{100} = 0,71$$

- Xác định hệ số phân phối mưa rào:

$$M_o = \frac{1}{1 + 0,001 \times F_v} = \frac{1}{1 + 0,001 \times 0,60} = 0,999$$

Tính lưu lượng Q_1, Q_2 tại đầu cống dọc ở G_1, G_2. Thời gian nước mưa tập trung trên mặt phủ t_1 = 5 phút.

Để tính t_r, t_c cần biết V_r, V_c. Bước đầu giả định V_r, V_c theo kinh nghiệm. Sau khi xác định trị số lưu lượng Q, độ dốc thiết kế cống i, các trị số V_r, V_c sẽ được kiểm tra ngược lại, nếu phù hợp với trị số giả định coi như đúng.

Ở ví dụ này, giả định V_r = 1.5 m/s, V_c = 1.7 m/s. Lưu lượng Q do F_1 gây ra xác định ở giếng thăm G_1 chỉ có t_1 và tr chảy ở $l = l_r$ = 80 m (lưu vực F_1)

t_r = 1,25 × 80/1,5 = 66s = 1 min

t = 5 + 1 = 6 min

$$Q_1 = M_o.Y.\ Fv.q = 0,999 \times 0,71 \times 0,60 \times \frac{5890}{(6+20)^{0,84}}$$

= 0,999 × 0,71 × 0,60 × 381,54

= 162,37 l/s = Q_{2-2}

Lưu lượng Q_{1-2} do F_1 gây ra ở giếng thăm G_2 có cả t_1, t_r, t_c. Do địa hình bằng phẳng cho r = 2. Tính được:

t_c = 2 × 80/1,70 = 94s = 2 min

t = 5 + 1 + 2 = 8 min

$$Q_{1\text{-}2} = 0,999 \times 0,71 \times 0,60 \times \frac{5890}{(8+20)^{0,84}}$$

= 0,999 × 0,71 × 0,60 × 358,51

= 152,57 l/s

Lưu lượng Q_{2-2} do F_2 gây ra ở G_2 chỉ có t_1 và t_r chảy ở $l = l_r = 80m$ dài lưu vực F_1, tính tương tự Q_1.

$$Q_{2-2} = 162,37 \text{ l/s}$$

Vậy lưu lượng Q_2 ở G_2 do cả F_1 và F_2 gây ra là:

$$Q_2 = Q_{1-2} + Q_{2-2} = 52,57 + 162,37 = 314,94 \text{ l/s}$$
$$= 0,315 \text{ m}^3/\text{s}$$

4) Lập chương trình tính lưu lượng nước mưa TN 4-1

Sau khi đã nắm vững được phương pháp tính bằng tay qua ví dụ 4-1 như trên, áp dụng công thức (4-13) và (4-9), ta dễ dàng lập được chương trình Turbo Pascal tính lưu lượng nước mưa như sau:

Program Tinh_lưu_lượng_nước_mưa_Qm;

```
ues crt;
var
        Ao,bo,C,m,n,P,t,t1,tr,tc,Vr,Vc,lr,lc,r: real ;
        Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y,Mo,Fv,q,t2,Qm: real ;
        i,nf: integer ;
        f: array[1.. 50] of integer ;
begin
clrscr ;
        write('nhập vào tần suất mưa P(năm):'); readln(P);
        write('nhập vào số Ao:'); readln(Ao);
        write('nhập vào số bo:'); readln(bo);
        write('nhập vào số C:'); readln(C);
        write('nhập vào số m:'); readln(m);
        write('nhập vào số n:'); readln(n);
        write('nhập vào chiều dài rãnh lr(m):'); readln(lr);
        write('nhập vào chiều dài cống lc(m):'); readln(lc);
        write('nhập vào tốc độ ở rãnh Vr(m/s):'); readln(Vr);
        write('nhập vào tốc độ ở cống Vc(m/s):'); readln(Vc);
        write('nhập vào diện tích lưu vực Fv(ha):'); readln(Fv);
        write('nhap vao so nf(%):'); readln(nf);
```

```
for i:=1 to nf do
     begin
        write('f[',i,']: ='); readln(f[i]);
            if f[i] = f[1] then Y1: =0,95;
            if f[i] = f[2] then Y2: =0,60;
            if f[i] = f[3] then Y3: =0,45;
            if f[i] = f[4] then Y4: =0,40;
            if f[i] = f[5] then Y5: =0,30;
            if f[i] = f[6] then Y6: =0,20;
            if f[i] = f[7] then Y7: =0,10;
     end ;
begin
            t1:=5;
            r:=2;
            tr:=(1.25*lr / Vr) / 60;
            tc:=(r*lc / Vc) / 60;
            t:=t1+tr+tc;
            Mo:=1/(1+0.001*Fv);
Y:=(f[1]*Y1+f[2]*Y2+f[3]*Y3+f[4]*Y4+f[5]*Y5+f[6]*Y6+f[7]*Y7)
/100;
            t2:=t+bo*exp(m*ln(P));
            q:=Ao*(1+C*(ln(P)/ln(10)))/exp(n*ln(t2));
            Qm:=q*Mo*Y*Fv;
     writeln('Hệ số phân phối mưa rào là Mo:', Mo:6:3);
     writeln('Hệ số dòng chảy là Y:',Y:6:3);
     writeln('Cường độ mưa rào là q:',q:8:2,' l/s.ha');
     writeln('Lưu lượng nước mưa là Qm:',Qm:8:2,' l/s');
end;
readln
end.
```

Chú ý: Đặc điểm chương trình TN 4-1 trên là:

- Dùng cấu trúc mảng *array* và *if ... then* để đưa toàn bộ bảng 4-3 vào chương trình.

- Vì bảng 4-3 chỉ có 7 giá trị Y1...Y7 nên nhập giá trị nf =7 .

- Khi máy đòi nhập f[i], theo ví dụ 4-1 sẽ nhập như sau:

f[1]: 70
f[2]: 0
f[3]: 0
f[4]: 0
f[5]: 8
f[6]: 0
f[7]: 22

- Công thức tính t có chia cho 60 để đổi đơn vị giây ra phút (vì tốc độ nước là m/s). Khác với tính tay thường quy tròn số phút cho đơn giản. Vì vậy kết quả tính toán có khác nhau chút ít.

Ví dụ nhập số liệu chạy để tính Q_{1-2} , Q_{2-2} rồi in ra được kết quả sau:

Qm = 154,33 l/s, (so với Q_{1-2} = 152,57 l/s)

Qm = 162,08 l/s, (so với Q_{2-2} = 162,37 l/s)

Cộng: *316,41 l/s* so với *314,94 l/s*

4-3 TÍNH TOÁN THUỶ LỰC MẠNG LƯỚI THOÁT NƯỚC ĐÔ THỊ. CHƯƠNG TRÌNH TN 4-2

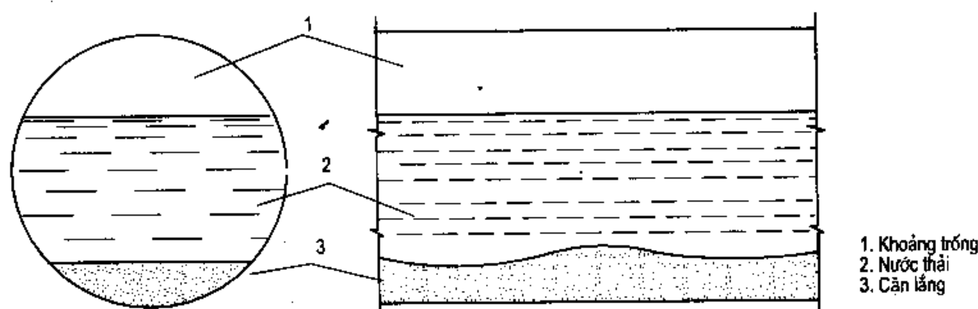
1) Đặc điểm chuyển động của nước thải đô thị

Nước thải đô thị thường có nhiều cặn lắng và rất khó lấy ra vì phức tạp và mất vệ sinh

Cặn lắng đọng lại trong cống thường chứa 3 – 8% (theo thể tích) là chất hữu cơ với kích thước > 1mm và 92 – 97% là cát. Trọng lượng riêng của cặn là 1,4 – 1,6 T/m^3.

Sơ đồ cấu trúc dòng chảy trong cống thoát nước xem hình 4-5 . Khi cửa xả bị ngăn bởi mức nước sông hồ, dòng chảy sẽ trở thành có áp và không còn khoảng trống.

Sơ đồ 4-5 cho ta thấy rõ: cặn lắng càng nhiều dòng chảy càng bị thu hẹp, giảm tốc độ nước chảy và càng ngày càng gây nguy cơ ngập úng trong đường đô thị.



Hình 4-5: *Sơ đồ cấu trúc dòng chảy*

Đặc trưng chuyển động của nước thải đô thị trong cống là hệ số Ray-nôn – R_e. Với cống tròn khi độ đầy hoàn toàn (h/d = 1), R_e được xác định theo công thức sau:

$$R_e = \frac{V.d}{\gamma} \qquad (4\text{-}15)$$

Trong đó: V - tốc độ nước trong cống, m/s;

d - đường kính cống;

γ - hệ số nhớt của nước thải.

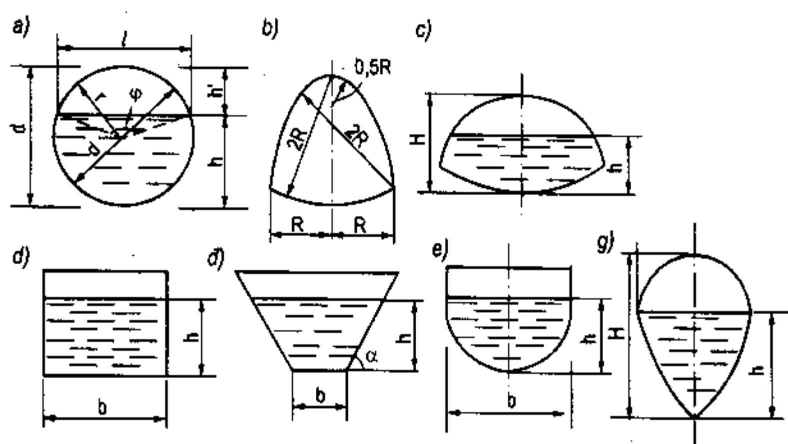
Trạng thái nước chảy trong cống, kênh mương đô thị có thể là chảy rối, chảy đều hoặc không đều, chảy ổn định hoặc không ổn định. Nhưng trong tính toán, với trường hợp có mặt thoáng, để đơn giản thường coi là chảy đều, không áp. Tức là coi tốc độ trung bình V = const, tiết diện chảy $\omega = F =$ const, lưu lượng q = const, độ dốc thuỷ lực I = độ dốc cống i, chu vi ướt $\chi = p_0 =$ const.

Còn trường hợp khác: do khu dân cư mở rộng cống cũ không đủ thoát nước, do cửa xả ra sông bị ngập, do bùn rác cản dòng chảy..., dẫn tới tình trạng cống chảy ngập có áp.

2) Các tiết diện cống và đặc tính thuỷ lực

Các loại tiết diện cống thoát nước giới thiệu trên hình 4-6. Thường dùng nhất là cống tròn. Kết cấu có thể là bê tông cốt thép, xây gạch, xây

đá... Tuy vậy, trong một công trình nên tránh dùng nhiều chủng loại cống khác nhau.



Hình 4-6: Các loại tiết diện cống thoát nước

Đặc tính thuỷ lực tốt nhất của tiết diện cống xác định bằng khả năng thoát nước lớn nhất, khi cùng có độ dốc và tiết diện ướt bằng nhau. Như vậy cống tròn là tốt nhất vì có bán kính thuỷ lực R_o lớn nhất.

$$R_o = \frac{\omega}{\chi} = \frac{F_o}{p_o} \qquad (4\text{-}16)$$

ω, F_o - diện tích ướt (m^2);

χ , p_o - chu vi ướt (m), là chu vi tiếp xúc giữa nước và thành cống;

a. Cống tròn

Với cống tròn đường kính d, khi nước chảy đầy một nửa và hoàn toàn có:

$$R_o = \frac{F_o}{p_o} = \frac{\pi.d^2}{4.\pi.d} = \frac{d}{4} = 0{,}25d \qquad (4\text{-}17)$$

Cống chảy (không áp) đạt tối đa khi $R_o = 0{,}304$ d, khi độ đầy

$a = h/d = 0{,}813$

Độ đầy a được định nghĩa là tỷ số h/d với cống tròn và h / H với các cống khác.

Với độ đầy không hoàn toàn ở cống tròn, h/d < 1, diện tích ướt F_o, chu vi ướt p_o có thể tính theo hình viên phân (khi h' < h):

$$F_o = \pi.r^2 - \frac{r(S-l)+l.h'}{2} \quad (4\text{-}18)$$

$$S = \varphi.\,r \quad (4\text{-}19)$$

r - bán kính cống;

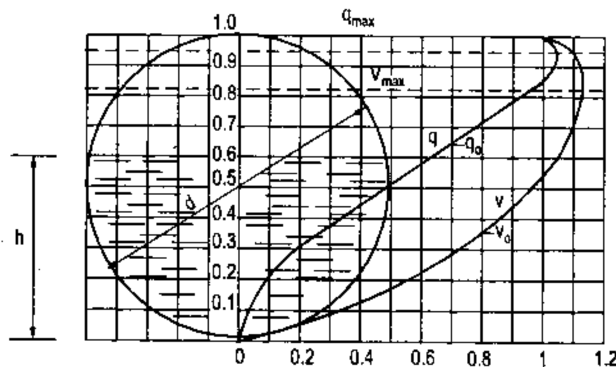
φ - góc chắn viên phân, tính bằng radian.

Góc tìm được theo công thức $\cos\varphi^o/2 = (r - h')/r$, tính ra độ (xem hình 4-6a).

$$p^o = \pi.\,d - S \quad (4\text{-}20)$$

Với cùng trị số R_o, tốc độ dòng chảy trong cống tròn khi độ đầy h/d = 0.5 và = 1 được xem là bằng nhau và tốc độ đạt tối đa khi h/d = 0.813. Lưu lượng thoát tối đa khi h/d = 0.95. Sau đó giảm dần, song lưu lượng khi nước chảy đầy hoàn toàn gấp đôi khi chảy một nửa.

Quan hệ giữa tốc độ, lưu lượng với độ đầy trong cống tròn giới thiêu trên hình 4-7. Trục đứng là tỷ lệ độ đầy h/d =0.1 ; 0.2 ; 0.3 ; ... ; 1.0. Trục ngang là mô đun tốc độ v_o và mô đun lưu lượng q_o, tính theo tỷ lệ đơn vị tốc độ và độ đầy khi chảy hoàn toàn.



Hình 4-7: *Các thành phần thuỷ lực của dòng chảy*

Gọi W_o và K_o là mô đun tốc độ và mô đun lưu lượng khi độ đầy h/d = 1. Nếu tính $C = (1/n).R^y$. Với y không đổi khi h thay đổi thì rõ ràng tỷ số $K/K_o = q_o$ và $W/W_o = v_o$ sẽ chỉ phụ thuộc vào độ nhám và kích thước của mặt cắt. Để tính nhanh, người ta tính sẵn K_o, W_o ứng với độ nhám n thường dùng.

Bảng 4-4 giới thiệu trị số K_o, W_o ứng với n = 0,013

Do cống tròn thoát nước tốt, dễ chế tạo hàng loạt và bền vững nên được dùng rộng rãi tới 90% trong mạng lưới thoát nước.

Bảng 4-4. Trị số K_o và W_o của cống tròn với n = 0,013

h = d (m)	0.25	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00
K_o m^3/s	0,595	3,76	11,2	24,0	43,7	71,0	106,5	152,5	208	276	354	447
W_o m/s	12,1	19,2	25,3	30,5	35,6	40,1	44,3	48,5	52,2	56,2	59,7	63,3

b. Kênh (cống) hình thang

Đặt độ dốc mái ta luy m = cotgα , chu vi ướt tính như sau:

$$p_o = b + 2h\sqrt{1 + m^2} \quad (4\text{-}21)$$

Diện tích mặt cắt ướt:

$$F_o = (b + mh)h = (b/h + m)h^2 \quad (4\text{-}22)$$

Để có mặt cắt có lợi nhất về thuỷ lực, cần có p_o bé nhất hay:

$$\frac{dp_o}{dh} = 0 \quad (4\text{-}23)$$

Với mặt cắt hình thang, tuỳ trị số m xác định được tỷ lệ β = b/h có lợi nhất về thuỷ lực theo bảng 4-5.

Bảng 4-5. Bảng trị số β = b/h có lợi nhất về thuỷ lực cho tiết diện hình thang

m	0	0,25	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50
β = b/h	2,00	1,562	1,236	1,00	0,828	0,702	0,606	0,532	0,472	0,424	0,385

Bảng 4-5 cho thấy, với mặt cắt chữ nhật m = 0, β =2 tức là có lợi nhất khi:

$$b = 2h$$

Bán kính thuỷ lực mặt cắt hình thang có lợi nhất khi $R_o = h/2$

Cần nhắc lại rằng: mặt cắt có lợi nhất về thuỷ lực không có nghĩa là lợi nhất về kinh tế – kỹ thuật, nhất là với kênh lớn. Nhưng với kênh nhỏ, ít đào sâu thì có thể có lợi cả về kinh tế kỹ thuật.

3) Công thức tính toán thuỷ lực mạng lưới cống

Tính toán đầu tiên ta thường gặp là tính lưu lượng.

Với đường ngoài đô thị, để tính lưu lượng chảy qua cầu cống, ta dựa vào địa hình, bản đồ... xác định các thông số tính toán như diện tích lưu vực, độ dốc lòng suối... rồi tính theo phương pháp cho trong quy trình. Quy trình này cũng có 2 loại: tính cho cống cầu nhỏ và cho cầu lớn.

Với đường đô thị, khi thiết kế mạng lưới thoát nước mưa, liên quan đến diện tích lưu vực đổ vào từng giếng thu hàm ếch, đó là những diện tích rất nhỏ (dưới 1 ha hoặc một vài ha) và được chia thành mảnh dọc 2 bên đường đô thị. Do vậy chịu ảnh hưởng chủ yếu bởi cường độ mưa, đặc điểm bề mặt phủ... như đã thể hiện qua tính toán ở mục 4-2.

Tuy vậy, giai đoạn tính toán lưu lượng nước mưa ở mục 4-2, ví dụ 4-1 có một điểm giống như tính lưu lượng cầu cống nói chung: đó là lưu lượng do thiên nhiên và địa hình tạo thành. Điểm khác nhau là: với đường ngoài đô thị, lưu lượng đó chảy qua một cầu hoặc cống. Với đường đô thị, lưu lượng đó tập hợp dần lại theo nhiều đoạn cống dọc, theo nhiều nhánh cống dọc để đổ ra cửa xả, Do đó, bị chi phối bởi một hệ thống các thông số tính toán: độ dốc lòng cống, đường kính, độ đầy... thay đổi theo từng đoạn cống, ảnh hưởng lẫn nhau. Lưu lượng sẽ được kiểm tra, khớp nối lại cho hài hoà với toàn hệ thống.

Công thức tính thuỷ lực mạng lưới cống đô thị, với giả định dòng chảy đều thường tính như sau:

Tính lưu lượng thoát nước của cống:

$$Q = F_o . V \qquad (4\text{-}24)$$

Để tính tốc độ, khi I = i, theo công thức Sê-di:

$$V = C\sqrt{R.i} \qquad (4\text{-}25)$$

Trong đó: Q - lưu lượng, m^3/s

F_o - diện tích tiết diện ướt, m^2;

V - tốc độ nước chảy, m/s;

R_o - bán kính thuỷ lực, m, xem công thức 4-16;

I - độ dốc thuỷ lực, lấy bằng độ dốc cống i ;

C - hệ số Sê-di. Theo Pavlốpki:

$$C = \frac{1}{n} R^y \qquad (4\text{-}26)$$

Trong đó:

n - hệ số nhám, độ nhám, phụ thuộc vật liệu và phương pháp chế tạo cống. Xem bảng 4-6.

y - số mũ, phụ thuộc độ nhám, hình dạng và kích thước của cống.

$$y = 2{,}5\sqrt{n} - 0{,}13 - 0{,}75\sqrt{R}(\sqrt{n} - 0{,}1) \qquad (4\text{-}27)$$

Khi d < 4000 mm thì n = 0,0013 và y = 1/6 = 0,1667

Các trị số ở bảng 4-6 tương ứng với điều kiện sản xuất hoàn chỉnh, nếu sản xuất bằng thủ công thì độ nhám sẽ lớn hơn nhiều. Ví dụ loại cống gang của ta sản xuất bằng khuôn cát, n = 0,02 – 0,03.

Bảng 4-6. Hệ số nhám n

Loại cống (kênh)	n trong công thức (4-26)
Cống	
- Sành	0,013
- Bê tông và bê tông cốt thép	0,014
- Xi măng amiăng	0,012
- Gang	0,013
- Thép	0,012
Kênh	
- Gạch	0,015
- Đá có trát vữa xi măng	0,017
- Đất	0,025

Gọi W là mô đun lưu tốc (hoặc đặc tính lưu tốc) khi i = 1

$$W = C\sqrt{R} \qquad (4\text{-}28)$$

Công thức Sê-di (4-25) viết thành:

$$V = W\sqrt{i} \qquad (4\text{-}29)$$

Gọi K là mô đun lưu lượng (hoặc đặc tính lưu lượng khi i = 1)

$$K = F_o C\sqrt{R} \qquad (4\text{-}30)$$

Công thức (4-24) viết thành:

$$Q = F_o.C\sqrt{Ri} = K\sqrt{i} \qquad (4\text{-}31)$$

Công thức (4-25), (4-31) là phương trình cơ bản của dòng chảy đều trong kênh hở (hoặc cống ngầm có khoảng trống).

Đối với trường hợp thường gặp là kênh (cống) hình thang, phương trình (4-31) nêu lên mối quan hệ giữa lưu lượng Q và bề rộng đáy b, chiều sâu nước h, độ dốc mái kênh ($m = \cot g\alpha$), độ dốc đáy i, độ nhám lòng kênh n .

$$Q = f(b, h, m, n, i) \qquad (4\text{-}32)$$

Với thoát nước đô thị, ta thường gặp các bài toán sau:

a) Đã biết lưu lượng (qua tính toán đã nêu ở mục 4-2, 4-3, ví dụ 4-1), xác định mặt cắt kênh (cống).

b) Với mạng lưới kênh (cống) đã có, tức là có mặt cắt hình học, do lưu vực thay đổi, (như cải tạo mở rộng khu dân cư), giải phương trình (4-32) gồm 6 biến số khi đã biết 5, còn lại 1 biến số lấy làm ẩn số.

c) Kiểm tra, xác định tình trạng ngập nước đô thị do cửa xả nước ra sông bị ngập, do lưu lượng vượt quá khả năng thoát nước của cống hiện có (và nhiều lý do khác nữa). Với bài toán này, chế độ chảy của cống ngầm trở thành có áp. Ta phải tính cao độ nước dâng, thời gian ngập nước... Bài toán khá phức tạp.

Với bài toán b), tìm h khi biết Q, b, m, n, i là một việc phức tạp nên phải giải bằng phương pháp thử dần. Ta cho các trị số h_1, h_2, h_3 ... tính ra các trị số F_o, C, R tương ứng, vẽ biểu đồ K – h (hình 4-8). Mặt khác tính ra $K_o = Q/\sqrt{i}$.

Trên biểu đồ K_o tìm ra trị số h_o tương ứng là trị số ta cần tìm.

Ví dụ 4-2: Cho một kênh thoát nước b = 4m, độ sâu h = 2m, m = 1.5, độ dốc i = 0,0002. Đây là kênh đất, sạch, thẳng, tra bảng xác định độ nhám n = 0,025. Tính lưu lượng Q của kênh và tốc độ nước V.

Trước hết tính bằng tay như sau:

$$F_o = (b + mh)h = (4 + 1.5 \times 2)2 = 14\ m^2$$

$$p_o = b + 2h\sqrt{1+m^2} = 4 + 2\sqrt{1+1,5^2} = 11,2\ m$$

$$R = F_o/p_o = 14/11.2 = 1,25\ m$$

$$C = \frac{1}{n}R^y$$

y theo công thức (4-27):

$$y = 2,5\sqrt{n} - 0,13 - 0,75\sqrt{R}(\sqrt{n} - 0,1)$$

$$= 2,5\sqrt{0,025} - 0,13 - 0,75\sqrt{1,25}(\sqrt{0,025} - 0,1) = 0,2163$$

$$C = \frac{1}{0,025} \times 1,25^{0,2163} = 41,98$$

Vậy $K = F_o.C\sqrt{R} = 14 \times 41,98\sqrt{1,25} = 657\ m^3/s$

$$Q = K\sqrt{i} = 657\sqrt{0,0002} = 9,3\ m^3/s$$

$$V = C\sqrt{R.i} = 41,98\sqrt{1,25 \times 0,0002} = 0,66\ m/s$$

4) Chương trình TN 4-2: Tương quan Q, V , h trong kênh, cống

Chương trình Pascal cho ví dụ 4-2 được lập như sau:

Program Tính_Q_kênh_đất_hình_thang;

```
uses crt;
var
        b, m, F, P, R, C, n, y, K, i, Q, V, h: real;
        h1, ho: integer;
begin
clrscr;
        write('nhap vao be rong day muong b(m):'); readln(b);
        write('nhap vao mai ta luy cotan(a) = m:'); readln(m);
        write('nhap vao do doc muong i:'); readln(i);
        write('nhap vao he so nham n:'); readln(n);
        write('nhap vao do sau muong h1(m):'); readln(h1);
```

```
for h1:=20 to ho do
  begin
        h:= h1/10;
        F:= (b + m*h)*h;
        P:=b + 2*h*sqrt(1 + sqr(m));
        R:= F/P;
        y:= 2.5*sqrt(n) – 0.13 – 0.75*sqrt(R)*(sqrt(n) – 0.1);
        C:= exp(y*ln(R))/n ;
        K:= F*C*sqrt(R);
        Q:= K*sqrt(i);
        V:= C*sqrt(R*i);
     writeln('Do sau muong la h:',h:6:2,' m');
     writeln('Mo dun luu luong K:',K:8:2,' m3/s');
     writeln('Luu luong cua kenh la Q:',Q:8:2,' m3/s');
     writeln('Toc do nuoc trong kenh la V:',V:8:2,' m/s');
  end;
readln
end.
```

Nhấn Ctrl+F9 rồi nhập số liệu lần lượt như ví dụ 4-2, sẽ ra kết quả K = 657 m^3/s, Q = 9,3 m^3/s, V = 0,66 m/s như trên. Chú ý nhập h1 = 20.

Lưu ý đặc điểm chương trình TN 4-2 là: cho trị số h1 thay đổi (tức h thay đổi) để vận dụng tìm chế độ chảy của kênh với Q khác nhau. Điều này sẽ thể hiện qua ví dụ 4-3.

Cho trị số h1 = 10h là để đưa h1 về số nguyên (integer), như vậy mới dùng lệnh ***for h1:=20 to ho do*** được .

Ví dụ 4-3: Cũng với kênh trên, nếu phải thoát với lưu lượng 15 m^3/s thì độ sâu h và tốc độ nước V là bao nhiêu ?

Tính bằng tay: trước hết tính:

$$K_o = \frac{Q}{\sqrt{i}} = \frac{15}{\sqrt{0,0002}} = 1060,7 \ m^3/s$$

Tính thử dần bằng giả định các trị số h, theo bảng 4-7 sau:

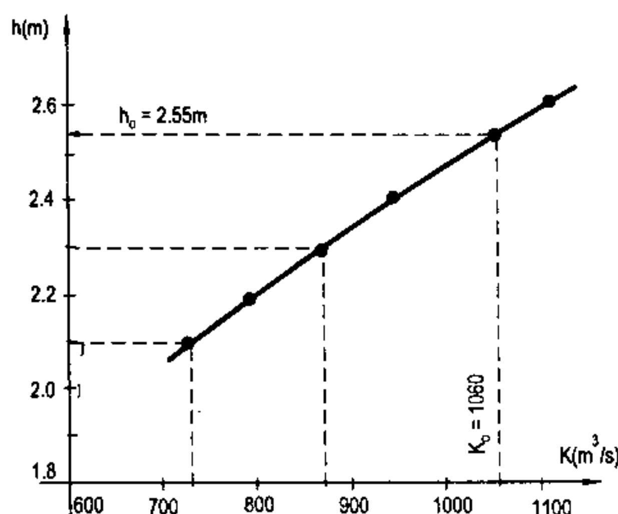
Bảng 4-7. Bảng tính tương quan h, K

h (m)	F_o (m^2)	p_o (m^2)	R (m^2)	C	K
2.1	15,015	11,572	1,30	42,39	725,7
2,2	16,060	11,932	1,35	42,74	797,5
2,3	17,135	12,293	1,39	43,03	869,3
2,4	18,240	12,653	1,44	43,36	949,1
2,5	19,375	13,014	1,49	43,69	1031,3
2,6	20,540	13,375	1,54	44,01	1121,8

Vẽ biểu đồ quan hệ K – h như hình 4-8, ứng với K_o = 1060,7 m^3/s ta xác định được h_o = 2,55 m

Tốc độ nước V có thể tính theo h = 2,5 m.

$$V = C\sqrt{R.i} = 43.69\sqrt{1,49 \times 0,0002} = 0,75\,m/s$$



Hình 4-8: *Biểu đồ quan hệ K – h*

Như vậy là mức nước kênh khi thoát với Q = 15 m^3/s, so với Q = 9,3 m^3/s đã nâng cao 2,55 – 2,00 = 0,55m.

Vận dụng chương trình Pascal TN 4-2 để giải bài toán ví dụ 4-3, sau khi nhập các số liệu, trong đó h1 = 26, ta được một dãy kết quả sau:

Do sau muong la h: 2.10 m

Mo dun luu luong K: 724.82 m3/s

Luu luong cua kenh la Q: 10.25 m3/s

Toc do nuoc trong kenh la V: 0.68 m3/s

Do sau muong la h: 2.20 m

Mo dun luu luong K: 796.02 m3/s

Luu luong cua kenh la Q: 11.26 m3/s

Toc do nuoc trong kenh la V: 0.70 m/s

Do sau muong la h: 2.30 m

Mo dun luu luong K: 871.04 m3/s

Luu luong cua kenh la Q: 12.32 m3/s

Toc do nuoc trong kenh la V: 0.72 m/s

Do sau muong la h: 2.40 m

Mo dun luu luong K: 949.96 m3/s

Luu luong cua kenh la Q: 13.43 m3/s

Toc do nuoc trong kenh la V: 0.74 m/s

Do sau muong la h: 2.50 m

Mo dun luu luong K: 1032.85 m3/s

Luu luong cua kenh la Q: 14.61 m3/s

Toc do nuoc trong kenh la V: 0.75 m/s

Do sau muong la h: 2.60 m

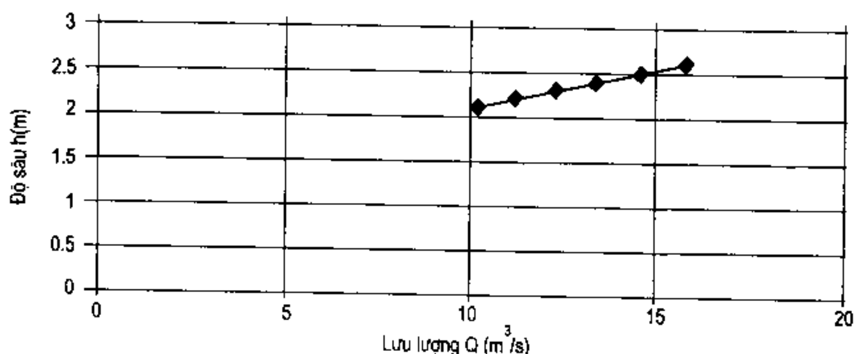
Mo dun luu luong K: 1119.78 m3/s

Luu luong cua kenh la Q: 15.84 m3/s

Toc do nuoc trong kenh la V: 0.77 m/s

Theo kết quả in ra như trên, ta có thể vẽ luôn biểu đồ quan hệ Q – h, rồi từ trị số Q = 15 m^3/s trên trục hoành, dóng lên sẽ tìm được h tương ứng là 2,55 m (hình 4-9)

Từ dãy kết quả trên, cũng xác định được tốc độ V = 0,75 m/s ứng với Q = 14,61 = 15 m^3/s.



Hình 4-9: Biểu đồ quan hệ Q – h.

4-4. CHƯƠNG TRÌNH TN 4-3: XÁC ĐỊNH ĐƯỜNG KÍNH CỐNG TRÒN THOÁT NƯỚC (VỚI H/D = 0.80) CHƯƠNG TRÌNH TN 4-4: TÌM V THEO Q VÀ D

1) Ví dụ 4-4

Xác định đường kính cống tròn thoát nước mưa ở ví dụ 4-1, có $Q = 0.316$ m^3/s, độ dốc $i = 0.006$, $n = 0,013$. Yêu cầu độ đầy $a = h/d \le 0,80$.

Nội dung tính toán bằng tay như sau:

Dựa vào biểu đồ hình 4-5, với a = 0.80, ta tra được $q_o = 1$, $q_o = K/K_o = 1$

$$K = Q/\sqrt{i}$$

Vậy:

$$K_o = K/q_o = \frac{Q}{q_o\sqrt{i}} = \frac{0,316}{1\times\sqrt{0,006}} = 4,08$$

Mặt khác theo công thức (4-30) khi chảy đầy ta có:

$$K_o = R_oC_o\sqrt{R_o} = \frac{\pi d^2}{4}\cdot\frac{1}{n}\left(\frac{d}{4}\right)^{1/6}\cdot\left(\frac{d}{4}\right)^{1/2}$$

Trong đó với cống tròn $R_o = d/4 = 0.25d$

Tương ứng:

$$C_o = \frac{1}{n}\cdot R_o^y = \frac{1}{0,013}\times 0,25^{1/6}d^{1/6} = 61,054d^{1/6}$$

$$\sqrt{R_o} = \sqrt{\frac{d}{4}} = \frac{d^{1/2}}{2}$$

$$F_o = \frac{\pi d^2}{4} = 0,7854.d^2$$

$$4,08 = 0,7854.d^2 \times 61,054 d^{1/6} \times \frac{d^{1/2}}{2} = 23,9759 d^{8/3}$$

$$d^{8/3} = 0.17017$$

$d = 0.51$ m, dùng $d = 0,50$ m

$h = 0.80\ d = 0,40$ m

Biểu đồ 4-7 với h/d =0.80 có $q_o = 1$

$$V_o = W/W_o = 1,18 = f_2(a)$$

Tra bảng 4-4 với d = 0.50 có $W_o = 19,2$

Vậy $V = W\sqrt{i}$ (công thức 4-29)

$$= V_o.\ W_o\sqrt{i} = 1,18 \times 19.2\ \sqrt{0,006}$$

$$= 1,75 \text{ m/s}$$

So với V_c giả định ở ví dụ 4-1 là 1,70 m/s xấp xỉ 1.75. Vậy đạt yêu cầu, không phải tính lại Q_2 ở ví dụ 4-1.

Giả sử, vì công trường chỉ có loại cống Ø75cm. Theo bảng 4-4 ta có $K_o = 11,2\ m^3/s$, $W_o = 25,3$ m/s

$$q_o = \frac{K}{K_o} = \frac{Q}{K_o\sqrt{i}} = \frac{0,316}{11,2\sqrt{0,006}} = 0,364$$

Tra biểu đồ 4-7 tìm được h/d = 0,40, $V_o = 0,90$

$$h_o = 0,40 \times 0,75 = 0,30 \text{ m}$$

$$V = W\sqrt{i} = V_o.W_o.\sqrt{i}$$

$$= 0,90 \times 25,5\sqrt{0,006} = 1,76 \text{m/s}$$

2) Lập chương trình TN 4-3

Đặc điểm chương trình này là:

- Cho trị số d biến đổi trong một khoảng nào đó, (phương pháp thử dần), trị số d cho Q xấp xỉ 0,316 m^3/s là trị số ta cần tìm.

- Ta vẫn kết hợp dùng biểu đồ 4-7 để tìm trị số qo. Ở ví dụ 4-4, vì h/d = 0.80 nên tra biểu đồ được $q_o = 1$.

Chương trình Pascal được lập như sau:

```
Program Tim_duong_kinh_cong_d_theo_Q;
uses crt;
var
    Q,qo,K,Ko,i,n,h,Fo,Ro,Co,y,V,Vo,W,d: real;
    d1,dn: integer;
begin
clrscr;
    write('nhap vao so tra bieu do 4-7 tri so qo:'); readln(qo);
    write('nhap vao do doc cong i:'); readln(i);
    write('nhap vao he so nham n:'): readln(n);
    write('nhap vao so mu y:'); readln(y);
    write('nhap vao duong kinh dn(m):'); readln(dn);
  for d1:=10 to dn do
    begin
     d:=d1/100;
     Fo:=pi*sqr(d)/4;
     Ro:=0.25*d;
     Co:=exp(y*ln(Ro))/n;
     Ko:=Fo*Co*sqrt(Ro);
     Q:=Ko*qo*sqrt(i);
     writeln('Duong kinh ong d(m):',d:6);
     writeln('Tri so Ko:',Ko:6:3);
     writeln('Tri so Co:',Co:6:3);
     writeln('Tri so Fo:',Fo:8:4);
     writeln('Luu luong Q(m3/s):',Q:8:3);
    end;
 readln
 end.
```

Nhập số liệu cho chạy, (khi n = 0,013 thì y = 1/6 = 0,1667):

qo: 1

i : 0,006

n : 0,013

y : 0,1667

dn: 52

được kết quả sau

d (m) =	0,49	0,50	0,51	0,52
Q (m^3/s) =	0,277	0,293	0,308	0,325

Vậy d = 0.515 m. Theo thiết kế cống định hình ta chọn d = 0.50 m.

3) Lập chương trình TN 4-4 tìm V theo Q và d

Để tiếp tục tìm tốc độ nước chảy V trong ví dụ 4-3, ta lập chương trình TN 4-4 sau:

```
            Program Tim_V_theo_Q_va_d;
    uses crt;
    var
            Q,qo,K,Ko,Fo,Ro,Co,y,n,V,Vo,Wo,d,i,ho,a: real;
    begin
    clrscr ;
            write('nhap vao luu luong Q(m3/s):'); readln(Q);
            write('nhap vao duong kinh cong d(m):'); readln(d);
            write('nhap vao do doc cong i:'); readln(i);
        begin
            n:=0.013;
            y:=0.1667;
            Fo:=pi*sqr(d)/4;
            Ro:=d/4;
            Co:=exp(y*ln(Ro))/n;
```

```
        Ko:=Fo*Co*sqrt(Ro);
        qo:=Q/(Ko*sqrt(i));
        Wo:=Co*sqrt(Ro):
        writeln('Tri so Wo la:',Wo:6:1);
        writeln('Tri so Ko:',Ko:6:2);
        writeln('Tri so qo:',qo:6:3,' m3/s');
        writeln('Từ qo tra biểu đồ 4-7 tìm a=ho/d và Vo:');
        writeln('Chiều sâu nước chảy là ho=a*d (m):');
        writeln('Tốc độ nước chảy là V=Vo*Wo*sqrt(i):');
    end;
readln
end.
```

Đặc điểm chương trình TN 4-4 là:

- Kết hợp với sử dụng biểu đồ 4-7.

- Bảng 4-4 đã được đưa vào chương trình qua áp dụng công thức $W_o = C_o\sqrt{R_o}$.

- Hàm số mũ $C_o = R_o^y / n$ đã áp dụng công thức (4-12) để đưa về exp

Cho chạy TN 4-4, nhập số liệu:

$Q(m^3/s) = 0,316$

d (m) = 0.75

i = 0.006

Ra kết quả: Trị số Wo là: 25,2

Trị số Ko: 11,13

Trị số qo: 0,366 m^3/s

Ta thấy kết quả tương quan Wo và Ko đúng như bảng 4-4 trên.

Từ qo tra biểu đồ 4-7 được độ đầy a = h/d = 0,40 và Vo = 0,90.

Vậy: ho = a*d = 0.40*0.75 = 0.30 m

V = Vo*Wo*sqrt(i) = 0.90*25.2*sqrt(0.006) = 1.76 m/s

Kết quả đúng như ví dụ 4-4 đã tính ở trên.

4-5. CHƯƠNG TRÌNH TN 4-5: TÍNH TỔN THẤT CỤC BỘ CỦA DÒNG CHẢY TRONG CỐNG.

1) Khái quát chung

Tổn thất cục bộ thường được tính ở giếng thăm, giếng chuyển bậc với cống có d > 500 mm và ở cửa xả ra kênh, sông thoát nước chảy qua đô thị. Do tổn thất cục bộ làm chậm tốc độ chảy, gây ra hiện tượng dềnh nước, làm lắng đọng bùn, lâu ngày có thể làm tắc cống.

Sơ đồ tổn thất cục bộ tại chỗ thu hẹp đột ngột và cửa ra thể hiện trên hình 4-10. Cột nước tổn thất cục bộ được tính theo công thức:

$$h_t = t_n \frac{V^2}{2g} \quad (4\text{-}33)$$

t_n - hệ số tổn thất cục bộ, xem bảng 4-8;

V (m/s) – tốc độ dòng chảy ở mặt cắt sau chỗ tổn thất, theo chiều nước chảy,

g = 9.8 m/s^2 - gia tốc trọng trường.

Bảng 4-8. Trị số t_n

Vị trí gây tổn thất cục bộ	t_n
- Cửa thu nước vào kênh mương	0,1
- Cửa thu nước vào ống gờ nhọn	0,5
- Cửa thu nước vào ống ở dưới mực nước	1,0
- Van khoá ở mức độ hở	
+ hoàn toàn	0,5
+ 3/4	0,26
+ 1/2	2,06
- Van ngược chiều	5
- Khuỷu ống 90°, ∅100 – 1000 mm	0.,39 - 0,5

Tại cửa xả là nơi mở rộng đột ngột, có thể tính cột nước theo định luật Booc-đa:

$$h_m = \frac{(V_1 - V_2)^2}{2g} \quad (4\text{-}34)$$

Áp dụng công thức (4-34) có thể kết hợp với đo đạc thực tế (như đo tốc độ V_2 của sông có nhiều cửa xả nước trong mùa ngập lụt)

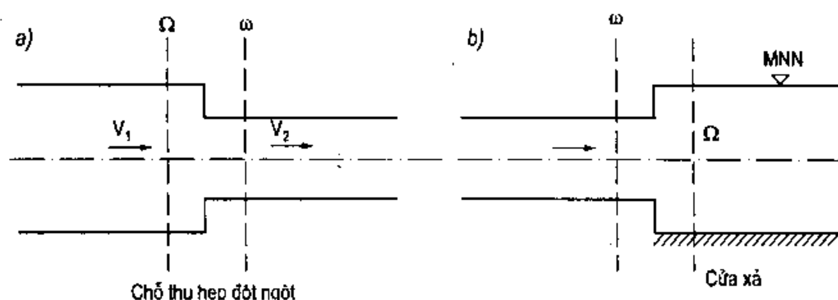
Hệ số tổn thất cục bộ t_n có thể tính cho trường hợp thu hẹp đột ngột (hình 4-10a):

$$t_b = 0,05\left(1 - \frac{f_n}{F_t}\right)^2 \qquad (4\text{-}35)$$

Trường hợp cửa xả:

$$t_x = \left(1 - \frac{f_n}{F_t}\right)^2 \qquad (4\text{-}36)$$

Trong đó f_n, F_t là diện tích mặt cắt nhỏ và to. Khi F_t lớn so với f_n thì $t_x = 1$.



Hình 4-10: *Một số dạng tổn thất cục bộ*

2) Ví dụ 4-5

Tính toán tổn thất và nước dềnh với sơ đồ hình 4-11. Trên cơ sở tính toán ở ví dụ 4-1, 4-2, 4-3, 4-4. Coi như cống ϕ500, ϕ750mm được đặt theo nguyên tắc độ đầy h/d = 0,8 (đường nét đứt) và mực nước thiết kế (MNTK) = 5,02m ở cao độ cửa xả, cũng tương ứng độ đầy 0,8 của cống ϕ750.

Tại G_1, $t_n = 0.50$, $V_1 = 1{,}75$ m/s. Tổn thất cột nước:

$$h_{t1} = 0{,}50 \times \frac{1{,}75^2}{2 \times 9{,}8} = 0{,}08\text{m}$$

Tại G_2, $t_n = 0.50$, $V_2 = 1.76$ m/s có:

$$h_{t2} = 0{,}50 \times \frac{1{,}76^2}{2 \times 9{,}8} = 0{,}08\text{m}$$

Tổn thất tại cửa xả, $t_n = 1$, V =0,75 m/s. (Tương ứng với kênh khi Q = 15 m^3/s).

$$h_{t3} = 1,00 \times \frac{0,75^2}{2 \times 9,8} = 0,03$$

Do mở rộng khu dân cư, mức nước tăng lên:

$$h_{t4} = 0,55m$$

(nên có mức nước thực tế MNTT = 5.02 + 0.55 = 5.57 m)

Vậy tổn thất cột nước toàn bộ là:

$$h_t = h_{t1} + h_{t2} + h_{t3} + h_{t4} = 0,74m$$

Vậy cao độ nước dềnh từ cao độ thiết kế 5.02 là:

$$h_d = 5,02 + 0,74 = 5,76 \text{ m}$$

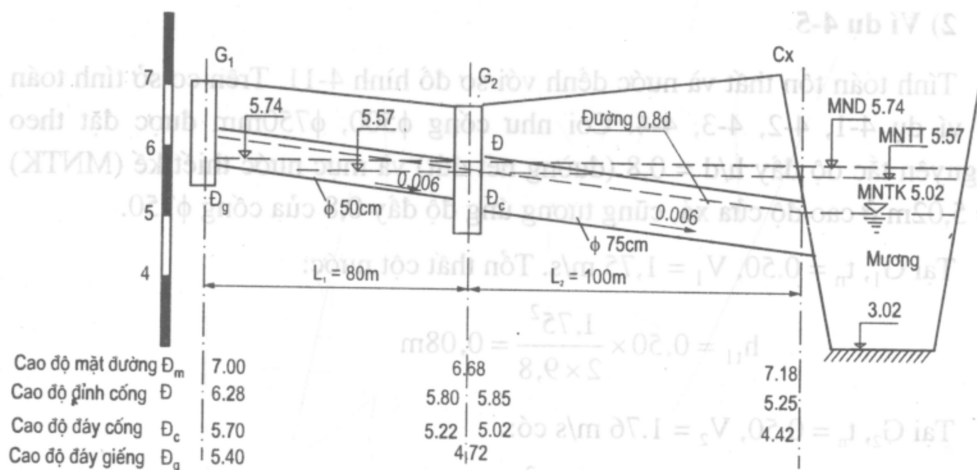
(Tính h_d đơn giản, coi mức nước dềnh bằng tổn thất cột nước)

Cao độ này ngập đến đỉnh cống ϕ750 ở G_2 và cống ϕ500 ở G_1 chỉ còn hở chiều cao là:

$$h' = 6,28 - 0,08 - 5,76 = 0,44 \text{ m}$$

(Trị số 0,08 là bề dày thành cống)

Như vậy, nguyên lý thoát nước cho F_1, F_2 lúc này là chảy theo xi phông trong đoạn $G_1 - C_x$.



Hình 4-11: Mặt cắt dọc cống theo ví dụ 4-5.

3) Lập chương trình TN 4-5: tính tổn thất cục bộ

```
              Program Tính_tổn_thất_cục_bộ;
ues crt;
var
        h,h1,h2,h3,h4,tn,tx: real;
        i,vf: integer;
        v: array[1..10] of integer;
begin
clrscr;
        write ('nhập vào mức nước dềnh cao hơn thiết kế h4(m):');
readln(h4);
        write('nhập vào hệ số tổn thất ở giếng thăm tn:'); readln(tn);
        write('nhập vào hệ số tổn thất ở cửa xả tx:'); readln(tx);
        write('nhập vào số vf:'); readln(vf);
   for i:=1 to vf do
     begin
        write('v[',i,']: ='); readln(v[i]);
        if v[i]=v[1] then h1: =tn*sqr(v[1]/100)/(2*9.8);
        if v[i]=v[2] then h2: =tn*sqr(v[2]/100)/(2*9.8);
        if v[i]=v[3] then h3: =tx*sqr(v[3]/100)/(2*9.8);
     end;
     begin
        h: = h1+h2+h3+h4;
        writeln('Tổn thất cột nước toàn bộ là h:',h:6:2,' m');
     end;
 readln
 end.
```

Chạy theo số liệu thí dụ 4-5, in ra kết quả như sau:

nhap vao muc nuoc denh cao hon thiet ke h4(m): 0.55

nhap vao he so ton that o gieng tham tn: 0.5

nhap vao he so ton that o cua xa tx: 1.0

nhap vao so vf: 3

v[1]: = 175

v[2]: = 176

v[3]: = 75

Ton that cot nuoc toan bo la h: 0.74 m

Đặc điểm chương trình TN 4-5 là:

- Dùng cấu trúc mảng *array* và *if...then* để đưa vào tốc độ nước ở các đoạn khác nhau

- Tốc độ V được nhân với 100 để thành số nguyên, dùng được lệnh *for i:=1 to vf do*

- Lưu ý: sau khi nhập vf = 3, ta phải nhập tiếp v[1]:=175, v[2]:=176, v[3]:=75 máy mới chạy ra kết quả h = 0.74 m.

4-6. MỘT VÀI VẤN ĐỀ VỀ PHẦN MỀM THOÁT NƯỚC

Tính toán thuỷ lực thoát nước đô thị là tính cho cả một mạng lưới với nhiều tuyến cống và các "nút" là các giếng thu – giếng thăm, rất phức tạp, nên cần xây dựng các phần mềm tính toán trên máy vi tính. Trong đó, trước hết là xây dựng các mô hình tính toán dựa trên các thông số cơ bản như: lượng mưa, mặt phủ, địa hình, địa chất, thuỷ văn, mật độ dân số, tiêu chuẩn cấp, thoát nước...

Từ những năm 1960 – 1970 các nước như Anh, Mỹ, Hà Lan, Đức, Canađa đã nghiên cứu và phát triển vấn đề này rất mạnh. Còn Việt Nam, xây dựng mô hình như thế nào để lập chương trình tính cho phù hợp với các thông số cơ bản của Việt Nam là vấn đề cần nghiên cứu, để việc tính toán thoát nước đô thị được nhanh chóng, hợp lý.

Trên tạp chí *Thế Giới mới* năm 2003 đã đưa một thông tin: có 20 nước ứng dụng công nghệ thông tin đạt hiệu quả nhất là: 1- Mỹ, 2- Singapore, sau đó đến các các nước khác như Anh, Pháp, Đức, Philipin, Thái Lan v.v... Đặc biệt không có cả Nga và Trung Quốc (?!), còn Việt Nam tất nhiên là "chưa có"!

Thời gian qua, công nghệ phần mềm của Việt Nam đã được quan tâm đầu tư và đã đạt được một số thành tựu, nhưng xem ra để được xếp hạng trong số 20 nước như Singapore, Philipin, Thái Lan thì còn lắm chông gai.

Phụ lục 1

TÓM TẮT CÁC THỦ TỤC VÀ HÀM CỦA TURBO PASCAL

1. Hàm ABS

Cú pháp:

```
ABS(r:Real): Real;
ABS(i:Integer):Integer;
```

Ví dụ:

```
Var a
Begin
        a:-10;Writeln(Abs(a):3);
End
```

Kết quả:

10

Nếu biến a là số thực (ví dụ -10,6) thì kết quả nhận được:

-10,6

2. Hàm ADDR

Cú pháp:

```
ADDR(Var <biến>):Pointer
```

Công dụng: Cho địa chỉ của biến.

3. Hàm APPEND

Cú pháp:

```
Append(Var F:Text);
```

Hàm Append dùng để mở tệp F và định vị con trỏ tại vị trí cuối của tệp.

4. Hàm ARC

Cú pháp:

```
Arc(x,y:Integer,gd,gc,R:Word);
```

Dùng để vẽ một cung tròn có tâm là (x, y), có bán kính R, góc đầu là gd, góc cuối là gc.

Ví dụ:

Arc(100,100,0,180,20);

5. Hàm ARCTAN

Cú pháp:

Arctan(r:Real):Real;

Cho giá trị dạng Real của Arctangent của đối số r.

Ví dụ:

Arctan(2.3)

Kết quả:

1.16

6. Thủ tục ASSIGN

Cú pháp:

Assign (F,<tên tệp>:string);

Dùng để gán tên tệp ghi trong <tên tệp> và biến F có dạng Text.

Ví dụ:

Assign(F,'CANBO');

Gán tệp 'CANBO' vào biến F, Trong đó biến F đã được khao báo có dạng Text.

7. ASSIGNCRT

Cú pháp:

AssignCrt(F)

Công dụng: Cho phép người sử dụng thay vì đưa kết quả ra màn hình lại đưa vào tệp F, Trong đó biến F đã được khai báo có dạng Text.

8. Thủ tục BAR (Graph nit)

Cú pháp:

Bar(x1,y1,x2,y2: Integer);

Dùng để vẽ một hình chữ nhật được xác định bởi góc trên bên trái có tọa độ (x1,y1) và góc dưới bên phải là (x2,y2).

Ví dụ:

Bar(10,20,100,200);

9. Thủ tục BAR3D (Graph Unit)

Cú pháp:

Bar3D(x1,y1,x2,y2:Integer, <sâu>: Word,Top:Boolean);

Thủ tục này dùng để vẽ một hộp ba chiều có tô màu, Trong đó bề mặt là hình chữ nhật được xác định bởi các tọa độ (x1,y1) và (x2,y2), chiều sâu xác định bởi biến <sâu>. Nếu biến Top là ON thì hình hộp có nắp, ngược lại, nếu biến Top là False thì hình hộp không có nắp.

Ví dụ:

Bar(50,50,100,100,60,On);

10. Thủ tục CHDIR

Cú pháp:

ChDir(s:String);

Dùng để thay đổi thư mục làm việc hiện thời sang thư mục có đường dẫn được chỉ thị trong chuỗi s.

Ví dụ:

ChDir('C:\HONG');

11. Hàm CHR

Cú pháp:

Chr(i:Integer);

Hàm Chr cho kết quả là kí tự có số thứ tự (mã ASCII) bằng I.

Ví dụ:

Chr(65);

Kết quả:

A

12. Thủ tục CIRCLE (Graph Unit)

Cú pháp:

Circle(x,y: Integer, R: Word);

Dùng để vẽ đường tròn có tâm là (x, y) và bán kính là R.

Ví dụ:

Circle(100,100,50);

13. Thủ tục CLEARDRIVE (Graph Unit)

Cú pháp:

ClearDrive

Dùng để xóa màn hình đồ họa.

14. Thủ tục CLEARVIEWPORT (Graph Unit)

Cú pháp:

ClearViewPort;

Dùng để xóa cửa sổ hiện thời được tạo ra bởi SetViewPort và được ghi nhận bằng GetViewSettings;

15. Thủ tục CLOSE

Cú pháp:

Close(Var F: File);

Trong đó biến F đã được khai báo dạng File. Thủ tục này dùng để đẩy vùng đệm của tệp F ra đĩa và đóng tệp.

Ví dụ:

Close(f);

16. Thủ tục CLOSEGRAPH (Graph Unit)

Cú pháp:

Closegraph;

Dùng để khôi phục màn hình trước khi khởi tạo đồ họa. Thủ tục này cũng giải phóng vùng nhớ do hệ đồ họa sử dụng. Hay nói một cách khác là đóng chế độ đồ họa.

17. Thủ tục CLREOL (Crt Unit)

Cú pháp:

ClrEol;

Dùng để xóa từ vị trí con trỏ đến cuối dòng hiện hành.

Ví dụ:

```
GotoXY(10,10);
ClrEol;
```

18. Thủ tục CLRSCR (Crt Unit)

Cú pháp:

ClrScr;

Dùng để xóa màn hình và đưa con trỏ về tọa độ (1,1).

19. Hàm CONCAT

Cú pháp:

Concat(S1,S2...Sn: String): String;

Hàm Concat dùng để tạo ra một chuỗi mới bằng cách nối các chuỗi là đối số của hàm: S1, S2, ..., Sn. Độ dài tối đa của chuỗi mới là 225. Nếu điều kiện này không thỏa mãn máy sẽ thông báo lỗi.

Ví dụ:

s1:='Hà Nội-'; s2:='Việt Nam';

s:=Concat(s1,s2);

Writeln(s);

Kết quả:

Hà Nội - Việt Nam

20. Hàm COPY

Cú pháp:

Copy(S:String; m,n: Integer): String;

Hàm cho kết quả là một chuỗi con được trích ra từ chuỗi S bắt đầu từ kí tự thứ m và lấy ra n ký tự.

Ví dụ:

s:='Nguyen Van Trinh';

Sc:=Copy(s,8,3);Writeln(se);

Kết quả:

Van

21. Hàm COS

Cú pháp:

Cos(X: Real): Real;

Hàm cho Cosine của góc X, Trong đó X được tính bằng Radian.

Ví dụ:

X:=0,5;Writeln(Cos(X):5:2);

Kết quả:

0.88

22. Thủ tục DELAY (Crt Unit)

Cú pháp:

Delay(tg: Word);

Tạm dừng chương trình trong khoảng thời gian tg, Trong đó tg được tính bằng miligiây.

Ví dụ:

Delay(1000);

23. Thủ tục DELETE

Cú pháp:

Delete(S: String;m,n: Integer);

Thủ tục Delete dùng để xóa đi m kí tự bắt đầu từ ký tự số n trong chuỗi S.

Ví dụ:

```
S:='Nguyen Van Trinh';
Sc:=Delete(S,8,4);
Writeln(Sc);
```

Kết quả:

Nguyen Trinh

24. DELLINE (Crt Unit)

Cú pháp:

Delline;

Xóa màn hình chứa con trỏ. Các dòng phía dưới sẽ cuộn lên một dòng.

25. Thủ tục DETECTGRAPH (Graph Unit)

Cú pháp:

DetectGraph(Var gd,gm: Integer);

Thủ tục này dùng để phát hiện kiểu màn hình và mode đồ họa và gán cho các biến gd, gm.

Ví dụ:

```
Uses Graph;
Var
    gd,gm: Integer;
Begin
    DetectGraph(gd,gm);
    Writeln("Kiểu màn hình:",gd:2);
    Witeln("Mode đồ họa:',gm:2);
End.
```

Kết quả (ví dụ):

Kiểu màn hình: 9
Mode đồ họa: 2

26. Hàm DISKFREE (Dos Unit)

Cú pháp:

DiskFree(<ổ đĩa>: Word); Longint;

Cho biết số byte còn trống trên ổ đĩa được chỉ định bằng biến <ổ đĩa>. Nếu <ổ đĩa> bằng:

+ 1 Cho kết quả của ổ A.

+ 2 Cho kết quả của ổ B.

+ 0 Cho ổ đĩa mặc định.

Ví dụ:

A:=DiskFree(0);

27. Hàm DISKSIZE (Dos Unit)

Cú pháp:

DiskSize(<ổ đĩa>: Word):Longint;

Cho biết dung lượng đĩa tính theo byte trên ổ đĩa được chỉ định bằng biến <ổ đĩa>. Nếu <ổ đĩa> bằng:

+ 1 Cho kết quả của ổ A.

+ 2 Cho kết quả của ổ B.

+ 0 Cho ổ đĩa mặc định.

Kết quả nhận được có dạng Longint.

Ví dụ:

A:=DiskSize(0);

28. Thủ tục DISPOSE

Cú pháp:

Dispose(P: Pointer);

Giải tỏa vùng bộ nhớ phân chia động cấp phát cho biến con trỏ P. Hàm này được dùng sau khi đã tạo ra một vùng cho biến con trỏ P bằng lệnh New.

29. Hàm DOSEXITCODE (Dos Unit)

Cú pháp:

Dosexitcode;

Trả về mã lỗi của tệp chương trình con. Kết quả có dữ liệu dạng Word. Kết quả nhận các giá trị như sau:

0	Chấm dứt bình thường.
1	Chấm dứt do ấn tổ hợp phím Ctrl+C.
2	Chấm dứt do lỗi thiết bị.
3	Chấm dứt do thủ tục Keep.

30. Thủ tục DOSVERSION (Dos Unit)

Cú pháp:

DosVersion;

Cho số liệu của hệ điều hành.

31. Thủ tục DRAWPOLY (Graph Unit)

Cú pháp:

DrawPoly(n,A);

Trong đó n có dạng Word và A là một mảng. Thủ tục này vẽ một đa giác có n đỉnh. Mảng A chứa tọa độ các đỉnh của đa giác.

32. Hàm EOF

Cú pháp:

Eof(F);

Trong đó F là một biến dạng File. Hàm Eof cho kết quả là một giá trị logic:

- True: Nếu con trỏ đặt ở cuối tệp.
- False: Nếu con trỏ không đặt ở cuối tệp.

33. Hàm EOLN

Cú pháp:

EoLn(F)

Trong đó F là biến dạng File. Hàm EoLn cho kết quả là một giá trị logic:

- True: Nếu con trỏ ở cuối dòng hoặc cuối tệp.
- False: Nếu con trỏ không ở cuối dòng hoặc cuối tệp.

34. Thủ tục ERASE

Cú pháp:

Erase(f);

Trong đó F là biến tệp. Hàm này dùng để xóa tệp được chứa trong tệp f.

Ví dụ:

```
Var
    f: File;
Begin
    Assign(f,'VIDU.PAS');
    Erase(f);
End.
```

Chương trình trên xóa tệp VIDU.PAS trong thư mục hiện thời.

35. Thủ tục EXEC

Cú pháp:

Exec(<đường dẫn>,<tham số>);

Hàm Exec thực hiện tệp có tên nằm trong <đường dẫn> với các tham số.

Ví dụ:

```
Exec('C:\BKED\BKED.EXE','Baocao');
```

36. Thủ tục EXIT

Cú pháp:

Exit;

Thủ tục Exit dùng để ra khỏi chương trình con hoặc dừng thực hiện chương trình.

37. Hàm EXP

Cú pháp:

Exp(X);

Trong đó X là số thực. Hàm cho giá trị có dạng số thực là kết quả của e mũ X.

Ví dụ:

```
Writeln(Exp(3):10:2);
```

Kết quả:

20.09

38. Hàm FEXPAND (Dos Unit)

Cú pháp:

FEXPand(P);

Trong đó P là một tên tệp. Hàm mở rộng tên tệp thành một tên tệp đầy đủ.

Ví dụ:

```
Uses Dos;
Var Sc: String;
Begin
    Sc:=FexPand('VIDU.PAS');
    Writeln(Sc);
End.
```

Kết quả (ví dụ):

C:\TURBO\VIDU.PAS

là tên đầy đủ bao gồm cả đường dẫn của tệp.

39. Hàm FILEPOS

Cú pháp:

FilePos(P);

Trong đó P là biến tệp. Hàm cho giá trị dạng Integer là số hiệu bản ghi (phần tử) mà con trỏ tệp F đang trỏ tới.

40. Hàm FILESIZE

Cú pháp:

FileSize(F);

Trong đó F là biến tệp. Hàm cho giá trị dạng Integer là số phần tử của biến tệp F.

41. Thủ tục FILLELLIPSE (Graph Unit)

Cú pháp:

FillEllipse(x,y,a,b);

Trong đó các biến: x, y dạng Integer, a, b có dạng Word. Thủ tục này vẽ và tô màu một hình ellipse có tâm là (x, y), bán kính dọc là a, bán kính ngang là b. Ellipse được tô bằng màu và mẫu hiện hành. Màu của đường viền là màu đang quy định bằng lệnh thủ tục SetColor.

Ví dụ:

FillEllipse(100,100,20,40);

42. Thủ tục FILLPOLY (Graph Unit)

Cú pháp:

FillPoly(n,A);

Trong đó n là biến dạng Word, A là một mảng. Thủ tục FillPoly tô màu cho đa giác gồm n đỉnh có các tọa độ đỉnh được chứa trong mảng A.

43. Thủ tục FLOODFILL (Graph Unit)

Cú pháp:

FloodFill(x,y,mau);

Trong đó biến x, y dạng Integer, biến mau có dạng Word. Thủ tục này dùng để tô một vùng khép kín trên màn hình đồ họa với màu vẽ và mẫu tô hiện hành. Tọa độ x, y phải được nằm trong vùng được tô.

Ví dụ:

FloodFill(100,100,Red);

44. Thủ tục FLUSH

Cú pháp:

```
Flush(F);
```

Trong đó F là biến tệp. Thủ tục Flush đẩy ra đĩa tất cả các kết quả đang nằm trong vùng đệm của tệp F.

45. Hàm FRAC

Cú pháp:

```
Frac(x);
```

Trong đó x là một số thực. Kết quả nhận được là một số thực phản ánh phần lẻ thập phân của x.

Ví dụ:

```
x:=12.34;y:=Frac(x);
Writeln(y:4:2);
```

Kết quả:

```
0.34
```

46. Thủ tục FREEMEM

Cú pháp:

```
Freemem(F,i);
```

Trong đó: F là biến con trỏ Pointer, i dạng Integer. Thủ tục Feemem giải phóng i byte bộ nhớ của vùng Heap liên quan đến biến F đã được cấp phát bằng thủ tục GetMem.

47. Thủ tục GETARCCOORDS (Graph Unit)

Cú pháp:

```
GetArcCoords(Var Arc Coord: ArcCoodsType)
```

Thủ tục này trả về các tọa độ đã được lệnh Arc hoặc Ellipe trước đó sử dụng. Cấu trúc của dạng dữ liệu ArcCoodsType như sau:

```
Type
    ArcCoordsType=Record
        x,y: Integer;
        XStart,YStart: Integer;
        XEnd,YEnd: Integer;
    End.
```

48. Thủ tục GETASPECTRATIO (Graph Unit)

Cú pháp:

Getaspectratio(X,Y);

Trong đó X, Y là các biến dạng Word. Hàm trên trả về cho biến X và Y độ phân giải của màn hình đồ họa theo chiều ngang (X) và chiều dọc (Y). Các bạn có thể tính được tỉ lệ của màn hình nhờ hai giá trị này.

49. Thủ tục GETBKCOLOR (Graph Unit)

Cú pháp:

GetBkColor;

Kết quả là dữ liệu dạng Word cho biết màu của nền hiện hành.

50. Thủ tục GETCOLOR (Graph Unit)

Cú pháp:

GetColor;

Kết quả là dữ liệu dạng Word cho biết màu của nét vẽ hiện hành.

51. Thủ tục GETDATE (Dos Unit)

Cú pháp:

GetDate(ngay,tháng,nam,ngay_tuan);

Hàm đưa vào các biến ngay, thang, nam, ngay_tuan dữ liệu dạng Word tương ứng với ngày, tháng, năm và ngày trong tuần của hệ thống.

52. Thủ tục GETDEFAULTPALETTE (Graph Unit)

Cú pháp:

GetDefaultPalette(Pa);

Trong đó Pa là biến có dạng PaletteType. Hàm đưa vào biến Pa bảng màu mặc định cho trình điều khiển đồ họa hiện hành.

Cấu trúc của PaletteType như sau:

```
Type
    PaletteType=Record
    Size;Byte;
    Colors: Array[0..MaxColor] of Shortint;
End.
```

53. Thủ tục GETDIR

Cú pháp:

GetDir(d,s);

Trong đó d dạng Byte và biến s dạng String. Hàm GetDir lấy thư mục cho ổ đĩa được chỉ định trong d và đưa vào biến s. Nếu d là 0 thì sẽ lấy trong ổ đĩa mặc định.

Ví dụ:

```
Var s:String;
Begin
    GetDir(0,s); Writeln(s);
End.
```

Kết quả (ví dụ):

```
C:\TURBO
```

54. GETDRIVERNAME (Graph Unit)

Cú pháp:

```
GetDriveName;
```

Kết quả là tên của trình điều khiển đồ họa dưới dạng một chuỗi kí tự (String).

55. Hàm GETENV

Cú pháp:

```
GetEnv(Env);
```

Trong đó Env là biến dạng String. Kết quả của hàm là đưa vào trong biến Env chuỗi môi trường cho biến môi trường.

56. Hàm GETATTR (Dos Unit)

Cú pháp:

```
GetAttr(F,thuoc_tinh);
```

Trong đó F là biến tệp (file) và thuoc_tinh là biến dạng Word. Kết quả của hàm là thuộc tính của tệp F. Trước khi thực hiện thủ tục này, biến F đã được gắn từ một tệp nhưng không được mở tệp.

Ví dụ:

```
Var thuoc_tinh: Word;
    F:File;
Begin
    Assign(F,'VIDU.PAS');
    GetAttr(F,thuoc_tinh);
    Writeln('Thuoc tinh: ',Thuoc_tinh);
End.
```

Kết quả (ví dụ):

```
Thuoc tinh: 32
```

57. Thủ tục GETFILLPATTERN (Graph Unit)

Cú pháp:

```
GetFillPattern(FP);
```

Trong đó FP là biến dạng FillPatternType. Hàm trên trả về trong biến FP định nghĩa của mẫu tô hiện hành. Cấu trúc của FillPatternType như sau:

```
Type
    FillPatternType=Array[1..8] of Byte;
```

58. Thủ tục GETFILLSETTiNG (Graph Unit)

Cú pháp:

```
GetFillSettings(FS);
```

Trong đó biến FS có dạng FillSettingsType. Hàm trên trả về cho biến FS màu và mẫu tô hiện hành. Cấu trúc của FillSettingsType như sau:

```
Type
    FillSettingsType=Record;
        Pattern: Word;
        Color: Word;
    End.
```

59. Thủ tục GETFTIME (Dos Unit)

Cú pháp:

```
GetFTime(F,tg);
```

Trong đó F là biến tệp, tg là biến dạng Longint. Hàm trên đưa vào biến tg thời gian của tệp F. Tệp F phải được gán và được mở trước khi dùng thủ tục này.

Ví dụ:

```
Var
    tg: Longint;
    f: File;
Begin
    Assign(F,'VIDU.PAS');
    Reset(f);GetFTime(f,tg);
    Writeln(tg);Close(f);
End.
```

60. Hàm GETGRAPHMODE (Graph Unit)

Cú pháp:

```
GetGraphMode;
```

Hàm trả về kiểu đồ họa là dữ liệu dạng Integer.

61. Thủ tục GETIMAGE (Graph Unit)

Cú pháp:

GetImage(x1,y1,x2,y2,B);

Trong đó các biến x1, y2, x2, y2 dạng Integer, biến B dạng Bitmap. Thủ tục trên cất hình chữ nhật (x1, y1, x2, y2) vào biến B để khi cần có thể gọi ra màn hình với nội dung đã cất nhưng ở các tọa độ khác nhau (sử dụng làm chuyển động các hình ảnh trong trò chơi hoặc trong phim hoạt hình).

62. Thủ tục GETLINESETTINGS (Graph Unit)

Cú pháp:

GetLineSettings(LS);

Trong đó LS có dạng LineSettingsType. Thủ tục trên đưa vào trong biến LS giá trị hiện hành của đường, mẫu, bề dày của hình ảnh. Cấu trúc của LineSettingsType như sau:

```
Type
    LineSettingsType=Record
        LineStyle: Word;
        Pattern: Word;
        Thickness: Word;
    End.
```

63. Hàm GETMAXCOLOR (Graph Unit)

Cú pháp:

GetMaxColor;

Kết quả là giá trị cao nhất cho màu trong bảng màu hiện hành thể hiện bằng dữ liệu dạng Word.

64. Hàm GETMAXMODE (Graph Unit)

Cú pháp:

GetMaxMode;

Kết quả là kiểu đồ họa có độ phân giải cao nhất của kiểu màn hình đang dùng.

65. Hàm GETMAXX (Graph Unit)

Cú pháp:

GetMaxX;

Hàm cho giá trị là tọa độ cao nhất theo chiều ngang của kiểu đồ họa hiện hành. Giá trị nhận được có kiểu dữ liệu dạng Integer.

66. Hàm GETMAXY (Graph Unit)

Cú pháp:

GetMaxY;

Hàm cho giá trị là tọa độ cao nhất theo chiềudọc của kiểu đồ họa hiện hành. Giá trị nhận được có kiểu dữ liệu dạng Integer.

67. Thủ tục GETMEM

Cú pháp:

GetMem(P,i);

Trong đó biến P dạng Pointer và biến i dạng Integer. Thủ tục trên cấp phát một vùng bộ nhớ i byte và lưu địa chỉ đầu của vùng nhớ này vào biến con trỏ P.

68. Hàm GETMODENAME (Graph Unit)

Cú pháp:

GetModeName(Mode);

Trong đó Mode là biến dạng Word. Hàm cho kết quả là một chuỗi (dạng của hàm là String) mô tả kiểu đồ họa ứng với mode đồ họa khai báo trong hàm.

69. Thủ tục GETMODERANGE (Graph Unit)

Cú pháp:

GetModeRange(dh,thap,cao);

Trong đó các biến dh, thap, cao có dạng Integer. Thủ tục đưa vào biến thap kiểu phân giải thấp nhất và biến cao kiểu phân giải cao nhất ứng với kiểu đồ họa dh.

70. Thủ tục GETPALETTE (Graph Unit)

Cú pháp:

GetPalette(P);

Trong đó P có dạng PaletteType. Thủ tục đưa vào biến P bảng màu hiện hành. Cấu trúc của PaletteType như sau:

```
Const
    MaxColors=15;
Type
    PaletteType=Record
        Size:Byte;
        Colors: Array[0..MaxColors] of Shortint;
    End;
```

71. Hàm GETPALETTESIZE (Graph Unit)

Cú pháp:

GetPaletteSize;

Hàm trả về số liệu lớn nhất của các bảng màu mà kiểu đồ họa hiện hành có thể cung cấp. Hàm có dữ liệu dạng Word.

72. Hàm GETPIXEL (Graph Unit)

Cú pháp:

GetPixel(x,y);

Trong đó x, y dạng Integer. Hàm trả về màu của điểm ảnh tại điểm (x, y).

73. Thủ tục GETTEXTSETTINGS (Graph Unit)

Cú pháp:

GetTextSettings(VB);

Trong đó VB có dạng GetTextSettingsType. Thủ tục đưa vào biến VB thiết kế văn bản hiện hành. Cấu trúc của GetTextSettingsType như sau:

```
Type
    GetTextSettingsType=Record
        Font: Word;
        Direction: Word;
        Charsize: Word;
        Horiz: Word;
        Vert: Word;
    End;
```

74. Thủ tục GETTIME (Dos Unit)

Cú pháp:

GetTime(Gio,Phut,Giay,PT_giay);

Trong đó các biến Gio, Phut, Giay, PT_giay là các biến dạng Word. Thủ tục GetTime đưa vào các biến trên tương ứng các giá trị về giờ, phút, giây và phần trăm giây của hệ thống.

75. Thủ tục GETVIEWSETTINGS (Graph Unit)

Cú pháp:

GetViewSettings(CS);

Trong đó CS có dạng ViewPortType. Thủ tục trên đưa vào trong biến VP thiết kế của cửa sổ hiện hành. Cấu trúc của ViewPortType như sau:

```
Type
    ViewPortTime=Record;
        x1,y1,x2,y2:Integer;
        Clip: Boolean;
    End;
```

76. Hàm GETX (Graph Unit)

Cú pháp:

GetX;

Hàm GetX đưa ra giá trị dạng Integer là tọa độ ngang của vị trí hiện hành.

77. Hàm GETY (Graph Unit)

Cú pháp:

GetY;

Hàm GetY đưa ra giá trị dạng Integer là tọa độ dọc của vị trí hiện hành.

78. Thủ tục GOTOXY (Dos Unit)

Cú pháp:

GotoXY(x,y);

Trong đó x, y là dữ liệu dạng Integer. Thủ tục này đưa con trỏ về vị trí (x, y) trên màn hình).

79. Thủ tục GRAPHDEFAULT (Graph Unit)

Cú pháp:

GraphDefault;

Thủ tục này thiết lập các giá trị ngầm định cho chế độ đồ họa.

80. Hàm GRAPHERRORMSG (Graph Unit)

Cú pháp:

GraphErrorMSG(<ma>);

Trong đó <ma> có dạng Integer. Hàm này cho kết quả dạng Integer thông báo lỗi ứng với <ma>.

81. Hàm GRAPHRESULT (Graph Unit)

Cú pháp:

GraphResult;

Hàm cho giá trị dạng Integer là mã lỗi của thủ tục đồ họa sau cùng.

82. Thủ tục HALT

Cú pháp:

Halt;

Dùng để kết thúc chương trình.

83. Hàm HI

Cú pháp:

Hi(i);

Trong đó i có dạng Integer. Kết quả của hàm dạng byte là Byte cao trong số nguyên i.

84. Thủ tục HIGHVIDEO (Crt Unit)

Cú pháp:

HighVideo;

Cho phép hiển thị màn hình với độ chói cao.

85. Hàm IMAGESIZE (Graph Unit)

Cú pháp:

ImageSize(z1,y1,x2,y2);

Trong đó: x1, y1, x2, y2 có dạng Integer. Hàm cho kết quả là số Byte cần thiết cất giữ trong Bitmap của phần màn hình được xác định bởi tọa độ (x1, y1, x2, y2).

86. Thủ tục INC

Cú pháp:

Inc(x[;n]);

Trong đó n dạng Longint. Thủ tục tăng giá trị của x lên n. Nếu không có n trong đối số của thủ tục thì tăng lên 1.

87. Thủ tục INITGRAPH (Graph Unit)

Cú pháp:

InitGraph(GraphDriver,GranphMode,Path);

Trong đó, GraphDrive và GraphMode dạng Integer, Path dạng String. Thủ tục khởi tạo môi trường đồ họa theo chương trình điều khiển GraphDriver và kiểu GraphMode. Nếu GraphDriver bằng 0 thì thủ tục tự động phát hiện bộ phối hợp màn hình và thiết lập kiểu có độ phân giải cao nhất. Thủ tục sẽ tìm các tệp .BGI theo đường dẫn được chỉ thị bởi Path.

Ví dụ:

Khởi động chế độ đồ họa khi các tệp .BGI nằm trong thư mục C:\TURBO\BGI:

```
Var
    gd,gm: Integer;
    s:String;
Begin
    gd:=0;
    s:='C:\TURBO\BGI';
    InitGraph(gd,gm,s);
End.
```

Các bạn có thể khởi động chế độ đồ họa trực tiếp như sau:

```
InitGraph(0,gm.'C:\TURBO\BGI');
```

88. Hàm INSERT

Cú pháp:

```
Insert(s1,s2,n);
```

Trong đó s1 và s2 dạng String, n dạng Integer. Hàm Insert chèn chuỗi s1 vào chuỗi s2 từ vị trí n.

Ví dụ:

```
Insert('hòa','Cộng xã',6);
```

Kết quả nhận được:

```
Cộng hòa xã
```

89. Thủ tục INSLINE (Crt Unit)

Cú pháp:

```
InsLine;
```

Thủ tục InsLine chèn dòng trống vào vị trí hiện hành của con trỏ trên màn hình.

90. Hàm INT

Cú pháp:

```
Int(x);
```

Trong đó x là số thực. Hàm cho kết quả là một số thực là phần nguyên của x.

Ví dụ:

```
Int(13.25);
```

Kết quả:

```
13
```

90. Hàm IORESULT

Cú pháp:

```
IOResult;
```

Kết quả của hàm có dạng Word là mã lỗi khi thực hiện thao tác vào/ra. Nếu IOResult khác 0 thì có nghĩa là đã gặp lỗi.

91. Thủ tục KEEP (Dos Unit)

Cú pháp:

```
Keep(ma);
```

Thủ tục kết thúc chương trình nhưng vẫn giữ nó trong bộ nhớ.

92. Hàm KeyPressed

Cú pháp:

KeyPressed;

Hàm cho kết quả là TRUE khi có một phím được ấn.

93. Hàm LENGTH

Cú pháp:

Length(s);

Trong đó s dạng String. Hàm cho kết quả dạng Integer là chiều dài của chuỗi s.

94. Thủ tục LINE (Graph Unit)

Cú pháp:

Line(x1,y1,x2,y2);

Trong đó x1, y1, x2, y2 là các giá trị dạng Integer. Thủ tục này kẻ một đường thẳng từ điểm (x1, y1) đến điểm (x2, y2).

95. Thủ tục LINEREL (Graph Unit)

Cú pháp:

LineRel(Dx,Dy);

Trong đó Dx, Dy có dạng Integer. Thủ tục trên kẻ một đường thẳng từ vị trí hiện tại của con trỏ (ví dụ x, y) đến điểm (x + Dx, y + Dy).

Ví dụ:

LineRel(100,50);

96. Thủ tục LINETO (Graph Unit)

Cú pháp:

LineTo(x,y);

Trong đó x, y có dạng Integer. Thủ tục kẻ một đường thẳng từ vị trí hiện thời của con trỏ đến điểm (x, y).

97. Hàm LN

Cú pháp:

Ln(x);

Trong đó x có dạng Real. Hàm cho một giá trị dạng Real là logarit tự nhiên của x.

98. Hàm LO

Cú pháp:

Lo(i);

Trong đó i có dạng Integer. Hàm cho giá trị dạng Byte là Byte thấp của số nguyên i.

99. Thủ tục LOWVIDEO

Cú pháp:

LowVideo;

Thủ tục LowVideo cho độ chói thấp của màn hình.

100. Thủ tục MARK

Cú pháp:

Mark(P);

Trong đó P có dạng Pointer. Thủ tục Mark đánh dấu địa chỉ đỉnh Heap trong con trỏ P.

101. Hàm MAXAVAIL

Cú pháp:

MaxAvail;

Hàm cho giá trị dạng LongInt là kích thước khối lớn nhất của vùng nhớ chưa cấp phát của bộ nhớ phân chia động.

102. Hàm MEMAVAIL

Cú pháp:

MemAvail;

Hàm cho giá trị dạng LongInt là tổng số vùng nhớ chưa cấp phát của bộ nhớ phân chia động.

103. Thủ tục MKDIR

Cú pháp:

MKDir(s);

Trong đó s dạng String. Thủ tục MKDir tạo một thư mục có tên chỉ định trong s.

Ví dụ:

MKDir('C:\TURBO\DULIEU');

hoặc:

s:= 'C:\TURBO\DULIEU';

MKDir(s);

104. Thủ tục MOVE

Cú pháp:

Move(v1,v2,i);

Trong đó v1, v2, i có dạng Integer. Thủ tục Move chép i Byte từ biến v1 sang biến v2.

105. Thủ tục MOVEREL (Graph Unit)

Cú pháp:

```
MoveRel(Dx,Dy);
```

Trong đó Dx, Dy có dạng Integer. Thủ tục MoveRel di chuyển con trỏ đến vị trí cách vị trí hiện hành Dx điểm ảnh theo chiều ngang và Dy điểm ảnh theo chiều dọc.

Ví dụ:

```
MoveRel(50,100);
```

106. Thủ tục MOVETO

Cú pháp:

```
MoveTo(x,y);
```

Trong đó x, y có dạng Integer. Thủ tục MoveTo định vị con trỏ tại tọa độ (x, y).

Ví dụ:

```
MoveTo(50,100);
```

hoặc:

```
x:=50, y:=100;
MoveTo(x,y);
```

107. Thủ tục NEW

Cú pháp:

```
New(P);
```

Trong đó P là biến con trỏ. Thủ tục New cấp phát vùng Heap cho biến con trỏ.

108. Thủ tục NORMVIDEO (Crt Unit)

Cú pháp:

```
NormVideo;
```

Khôi phục thuộc tính màn hình mặc định theo các thuộc tính đã có ở vị trí con trỏ khi thực hiện chương trình.

109. Thủ tục NOSOUND

Cú pháp:

```
NoSound;
```

Dùng để tắt âm thanh đã được tạo ra bởi thủ tục Sound.

110. Hàm ODD

Cú pháp:

```
Odd(i);
```

Trong đó i có dạng Integer. Hàm cho giá trị True khi i lẻ và cho giá trị False khi i chẵn.

111. Hàm ORD

Cú pháp:

```
Ord(s);
```

Hàm cho số thứ tự của s trong kiểu vô hướng đếm được. Thứ tự được đánh số từ 0 cho đến n-1 (n là số phần tử).

Ví dụ:

```
Type
    ds=(MOT,HAI,BA,BON);
Var
    h:ds;
Begin
    Writeln(Ord(HAI):2);
End.
```

Kết quả nhận được:

```
1
```

112. Thủ tục OUTTEXT (Graph Unit)

Cú pháp:

```
OutText(s);
```

Hiển thị tại vị trí con trỏ chuỗi s theo các tham số về phông chữ, kích thước... được thiết lập bởi SetTextStyle và được thiết lập bởi SetColor.

113. Thủ tục OUTTEXTXY (Graph Unit)

Cú pháp:

```
OutTextXY(s,y,s);
```

Trong đó x, y dạng Integer. Hiển thị chuỗi s tại tọa độ (x, y) theo các tham số đã được thiết lập bởi các hàm và thủ tục liên quan.

Ví dụ:

```
OutTextXY(100,100,"Nho Hen');
```

114. Hàm PI

Cú pháp:

```
PI;
```

Cho kết quả là hàm số Pi.

115. Thủ tục PIESLINE (Graph Unit)

Cú pháp:

```
PiesLine(x,y,gd,gc,r);
```

Trong đó x, y dạng Integer các giá trị gd, gc, r dạng Word. Thủ tục PiesLine vẽ một hình quạt có tâm là (x, y), bán kính r và được vẽ từ góc đầu gd đến góc cuối gc.

116. Hàm POS

Cú pháp:

Pos(s1,s2);

Cho kết quả là vị trí của chuỗi s1 trong chuỗi s2. Nếu s1 không nằm trong s2 thì hàm cho giá trị 0.

Ví dụ:

Pos('hoa',"Cong hoa');

sẽ cho kết quả:

6

117. Hàm PRED

Cú pháp:

Pred(s);

Cho kết quả dạng Integer là vị trí trước của s trong kiểu dữ liệu vô hướng đếm được.

118. Thủ tục PUTIMAGE (Graph Unit)

Cú pháp:

PutImage(x,y,B,ht);

Trong đó x, y dạng Integer, B là biến Bitmap, ht chỉ định cách hiển thị. Thủ tục này cho hiển thị nội dung của biến Bitmap B bắt đầu tại điểm (x, y). Cách hiển thị nhận các giá trị sau:

Mã	Giá trị số
CopyPut	0
XORPut	1
ORPut	2
ANDPut	3
NOTPut	4

Ví dụ:

PutImage(50,100,B,CopyPut);

Trong đó biến Bitmap B đã được định nghĩa bởi thủ tục GetImange.

119. Thủ tục PUTPIXEL (Graph Unit)

Cú pháp:

PutPixel(x,y,mau);

Trong đó x, y dạng Integer, mau quy định mã màu (hoặc giá trị số). Thủ tục PutPixel vẽ một điểm tại tọa độ (x, y) với màu do biến mau quy định.

Ví dụ:

PutPixel(100,100,Red);

120. Hàm RANDOM

Cú pháp:

Random(i);

Random;

Kết quả là một số ngẫu nhiên do Turbo Pascal tạo ra. Nếu bạn chỉ thị:

Random(i);

Trong đó i là một số dạng Word thì tạo ra một số nguyên từ 0 đến i-1. Nếu bạn chỉ thị:

Random;

(không có đối số) thì kết quả là một số thực không âm và nhỏ hơn 1.

Ví dụ:

Random(10);

Kết quả nhận được là một số nguyên ngẫu nhiên từ 0 đến 9.

121. Thủ tục READ và READLN

Cú pháp:

Read(<biến>);

Readln(<biến>);

Đọc số liệu được nhập từ bàn phím hay từ tệp dữ liệu.

122. Hàm READKEY

Cú pháp:

ReadKey;

Kết quả có dạng Char. Hàm ReadKey dùng để đọc một kí tự từ bàn phím. Nếu kết quả bằng 0 nghĩa là đã ấn một phím đặc biệt như Ctrl, End...

123. Thủ tục RECTANGLE

Cú pháp:

Rectangle(x1,y1,x2,y2);

Trong đó x1, y1, x2, y2 là các giá trị dạng Integer. Thủ tục trên dùng để kẻ một hình chữ nhật được xác định bởi góc trên bên trái là (x1, y1) và góc dưới bên phải là (x2, y2).

Ví dụ:

```
Rectangle(50,70,100,200);
```

124. Hàm REGISTERBGIDRIVER (Graph Unit)

Cú pháp:

```
RegisterBGIDriver(D);
```

Trong đó D có dạng Pointer. Kết quả của hàm là một giá trị Integer dùng để nạp tệp chứa chương trình điều khiển .BGI và đăng kí chương trình điều khiển với hệ đồ họa. D là con trỏ đến vị trí của chương trình điều khiển .BGI. Nếu có lỗi thì hàm trả về giá trị âm, còn ngược lại thì hàm trả về số hiệu chương trình điều khiển đã gắn.

125. Hàm REGISTERBGIFONT (Graph Unit)

Cú pháp:

```
RegisterBGIFont(font);
```

Trong đó font dạng Pointer. Kết quả của hàm dạng Integer dùng nạp phông .BGI và đăng kí phông với hệ đồ họa. Nếu có lỗi thì hàm cho giá trị âm. Ngược lại, hàm cho giá trị là số hiệu phông đã đăng kí.

126. Thủ tục RELEASE

Cú pháp:

```
Release(P);
```

Trong đó P là biến có dạng Pointer. Hàm giải tỏa vùng nhớ Heap đã cấp phát tính từ Mark gần nhất trước đó. P cất giữ địa chỉ của đỉnh Heap.

127. Thủ tục RENAME

Cú pháp:

```
Rename(F,s);
```

Trong đó F là biến dạng File, s có dạng String. Thủ tục Rename đổi tên tệp F thành s.

Ví dụ:

```
Var
    F:File;
    s:String;
Begin
    Assing(f,'VIDU.PAS');
    Rename(f,'LUU.PAS');
End.
```

Chương trình trên đổi tệp VIDU.PAS thành tệp mới có tên là LUU.PAS.

128. Thủ tục RESET

Cú pháp:

Reset(f);

Trong đó f là biến File. Thủ tục Reset dùng để mở tệp có biến tệp là f và chỉ đọc.

129. Thủ tục RESTORECRTMODE (Graph Unit)

Cú pháp:

RestoreCRTmode;

Dùng để khôi phục kiểu làm việc của màn hình trước khi khởi tạo chế độ đồ họa.

130. Thủ tục REWTITE

Cú pháp:

Rewrite(f,[;i];

Trong đó f là biến tệp và i có dạng Integer. Thủ tục này chuẩn bị tệp để ghi. Nếu tệp này chưa tồn tại thì Turbo Pascal sẽ tạo ra. Nếu là tệp không kiểu thì i sẽ xác định chiều dài của một bản ghi.

131. Thủ tục RMDIR (Dos Unit)

Cú pháp:

RmDir(s);

Trong đó s có dạng String. Thủ tục này dùng để hủy thư mục được chỉ định trong s.

132. Hàm ROUND

Cú pháp:

Round(x);

Trong đó x có dạng Real. Hàm Round dùng để làm tròn x và kết quả nhận được có dạng LongInt.

Ví dụ.

Round(12.52);

Kết quả:

13

và kết quả của hàm:

Round(13.31);

là:

12

133. Thủ tục SECTOR (Graph Unit)

Cú pháp:

Sector(x,y,gd,gc,n,d);

Trong đó x, y có dạng Integer, gd, gc, n, d dạng Word. Thủ tục Sector dùng để vẽ một cung tròn có các tham số như sau:

+ Tâm là (x, y).

+ Bán kính ngang là n.

+ Bán kính dọc là d.

+ Đi từ góc gd.

+ Kết thúc ở góc gc.

Ví dụ:

Sector(100,100,0.180,40,60);

134. Thủ tục SEEK

Cú pháp:

Seek(F,P);

Trong đó F là biến tệp (file) và P có dạng Integer. Thủ tục này dùng để di chuyển con trỏ tệp đến điểm đầu của bản ghi thứ P trong tệp có biến tệp F.

Ví dụ:

Assign(F,'VD.TXT');

Seek(F,10);

135. Thủ tục SEEKEOF

Cú pháp:

SeekEof(F);

Trong đó F là biến tệp. Kết quả có dạng Boolean. Hàm cho giá trị True nếu con trỏ ở vị trí cuối tệp, ngược lại cho giá trị False. Trước khi kiểm tra dấu hiệu kết thúc tệp, hàm SeeEof nhảy qua khoảng trống và Tab (khác với hàm Eof ở điểm này).

136. Thủ tục SEEKEOLN

Cú pháp:

SeekEoln(F);

Trong đó F là biến tệp. Hàm này cho giá trị True nếu con trỏ ở vị trí cuối dòng, ngược lại cho giá trị False.

137. Hàm SEG

Cú pháp:

Seg(<biến>);

Seg(<Procedure> hay <Function>);

Kết quả của hàm có dạng Word là giá trị của biến, thủ tục hay hàm trong chương trình.

138. Thủ tục SETACTIVEPAGE (Graph Unit)

Cú pháp:

SetActivePage(tr);

Trong đó tr có dạng Word. Thủ tục này dùng để chọn màn hình đồ họa làm trang hoạt động.

139. Thủ tục SETALLPALETTE (Graph Unit)

Cú pháp:

SetAllPalette(Pa);

Trong đó Pa là biến Palette. Thủ tục này thay đổi bảng màu theo định nghĩa trong Palette.

140. Thủ tục SETASPECTRATIO (Graph Unit)

Cú pháp:

SetAspectRatio(x,y);

Trong đó x, y có dạng Word. Thủ tục dùng để hiển thị đồ họa theo tỉ số x chia cho y.

141. Thủ tục SETBKCOLOR (Graph Unit)

Cú pháp:

SetBkColor(ma);

Trong đó ma là mã màu (hoặc giá trị số) có dạng Word. Thủ tục này dùng để thiết lập màu nền cho chế độ đồ họa theo bảng màu hiện hành.

Ví dụ:

SetBkColor(Blue);

142. Thủ tục SETDATE (Dos Unit)

Cú pháp:

SetDate(ngay,thang,nam);

Trong đó, ngay, thang, nam có dạng Word. Thủ tục này dùng để thiết lập lại đồng hồ hệ thống theo ngày, tháng, năm truyền qua tham biến tương ứng trong đối số của hàm.

143. Thủ tục SETFETTR (Dos Unit)

Cú pháp:

SetFattr(F,At);

Trong đó F là biến tệp (file) và At có dạng Word. Thủ tục này lập thuộc tính của tệp F thành At. Tệp F được gán nhưng không được mở trước khi dùng thủ tục này.

144. Thủ tục SETFILLPATTERN (Graph Unit)

Cú pháp:

SetFillPattern(Pa,Color);

Trong đó Pa có dạng FillPatternType, Color có dạng Word. Thủ tục dùng để thiết lập mẫu tô do người dùng định nghĩa.

145. Thủ tục SETFILLSTYPE (Graph Unit)

Cú pháp:

SetFillStyle(Pattern,Color);

Thủ tục này ấn định mẫu tô của màu tô.

146. Thủ tục SETFTIME (Dos Unit)

Cú pháp:

SetFTime(F,Time);

Trong đó F là biến tệp, Time có dạng LongInt. Thủ tục này dùng để thiết lập thời gian cho tệp F theo giá trị trong Time.

147. Thủ tục SETGRAPHBUFSIZE (Graph Unit)

Cú pháp:

SetGraphBufSize(Buf);

Trong đó Buf dạng Word. Thủ tục này dùng để ấn định kích cỡ vùng đệm đồ họa.

148. Thủ tục SETGRAPHMODE (Graph Unit)

Cú pháp:

SetGraphMode(mode);

Trong đó mode có dạng Integer. Dùng để thiết lập kiểu đồ họa hiện hành.

149. Thủ tục SETINTVEC (Dos Unit)

Cú pháp:

SetIntVec(Int,Vec);

Trong đó Int dạng Byte và Vec dạng Pointer. Thủ tục này dùng để cất giữ véctơ ngắt số Int vào địa chỉ Vec.

150. Thủ tục SETLINESTYLE (Graph Unit)

Cú pháp:

SetLineStyle(dạng,mẫu,bề dày);

Trong đó các giá trị: dạng, mẫu, bề dày có dạng Word. Thủ tục này ấn định dạng, mẫu, bề dày của đường vẽ.

151. Thủ tục SETPALETTE (Graph Unit)

Cú pháp:

SetPalette(mau1,mau2);

Trong đó mau1, mau2 có dạng Word. Thủ tục thay chỉ số mau1 bằng mau2 trong bảng màu.

152. Thủ tục SETRGBPalette (Graph Unit)

Cú pháp:

SetGRBPalette(mau1, Mã màu đỏ, Mã màu xanh, Mã màu lục);

Màu có chỉ số mau1 trong bảng màu được thay bằng các màu đỏ, xanh, lục.

153. Thủ tục SETTEXTBUF

Cú pháp:

SetTextBuf(f,Buf);

SetTextBuf(f,Buf,kt);

Trong đó f là biến tệp, kt có dạng Word. Thủ tục dùng để gán tệp văn bản f vào vùng đệm của Buffer. Kích thước mặc định của vùng đệm sẽ bị thay đổi nếu có chỉ thị kt (dạng hai).

154. Thủ tục SETTEXTJUSTIFY (Graph Unit)

Cú pháp:

SetTextJustify(Horiz,Vert);

Xác định vị trí hiển thị văn bản trong màn hình đồ họa.

155. Thủ tục SETTEXTSTYLE (Graph Unit)

Cú pháp:

SetTextStyle(phông,chiều,kích_thước);

Dùng để thiết lập phông chữ, chiều hiển thị (0=ngang, 1=dọc và kích thước của chữ trong màn hình đồ họa).

156. Thủ tục SETTIME (Dos Unit)

Cú pháp:

SetTime(giờ,phút,giây,PT_giây);

Các tham số của thủ tục có dạng Word. Thủ tục dùng để thiết lập thời gian hệ thống bao gồm giờ, phút, giây, và phần trăm giây.

157. Thủ tục SETUSERCHARSIZE (Graph Unit)

Cú pháp:

SetUserCharSize(x1,x2,y1,y2);

Các tham số của thủ tục có dạng Word. Thủ tục này dùng để thay đổi tỉ lệ hiển thị của văn bản trong chế độ đồ họa theo chiều rộng và chiều cao.

Ví dụ:

SetUserCharSize(1,3,1,2);

Kết quả là chiều rộng bằng 1/3 bề rộng bình thường và chiều cao bằng 1/2 chiều cao bình thường.

158. Thủ tục SETVIEWPORT (Graph Unit)

Cú pháp:

SetViewPort(x1,y1,x2,y2,Clip);

Trong đó x1, y1, x2, y2 có dạng Integer, Clip nhận hai giá trị ClipON và ClipOFF. Thủ tục chọn hình chữ nhật có đỉnh trên bên trái là (x1, y1) và tọa độ đỉnh dưới bên phải là (x2, y2) làm cửa sổ làm việc. Tọa độ của các điểm tuân theo cửa sổ mới này. Nếu chọn ClipOn thì cho phép vẽ các điểm ra ngoài cửa sổ, ngược lại thì các điểm chỉ được vẽ bên trong cửa sổ.

Ví dụ:

SetViewPort(100,50,200,300,ClipOn);

159. Thủ tục SETVISUALPAGE (Graph Unit)

Cú pháp:

SetVisualPage(tr);

Trong đó tr có dạng Word. Thủ tục chọn trang đồ thị để hiển thị lên màn hình.

160. Thủ tục SETWRITEMODE (Graph Unit)

Cú pháp:

SetWriteMode(mode);

Chọn một trong hai kiểu đường: CopyPut (0) hoặc XORPut (1).

161. Hàm SIN

Cú pháp:

Sin(x)

Hàm cho giá trị Sin(x).

162. Hàm SIZEOF

Cú pháp:

SizeOf(<biểu thức>);

Hàm cho giá trị dạng Word là số Byte của biến hay kiểu dữ liệu được ghi trong <biểu thức>.

163. Thủ tục SOUND

Cú pháp:

Sound(<bt số>);

Phát ra âm thanh có tần số là <bt số>.

Ví dụ:

Sound(300);

164. Hàm SPTR

Cú pháp:

Sptr;

Hàm trả về giá trị hiện hành của thanh ghi S.

165. Hàm SQR

Cú pháp:

Sqr(x);

Hàm cho giá trị bình phương của x. Nếu x dạng Real thì kết quả của hàm dạng Real. Nếu x dạng Integer thì kết quả nhận được cũng có dạng Integer.

166. Hàm SQRT

Cú pháp:

Sqrt(x);

Hàm cho kết quả là căn bậc hai của x. Trong đó x có thể là số nguyên hoặc số thực nhưng kết quả của hàm luôn là số thực.

167. Hàm SSEG

Cú pháp:

SSeg;

Hàm cho giá trị của thanh ghi SS.

168. Thủ tục STR

Cú pháp:

Str(i,[:n],s);

Str(r,[:n:[tp],s];

Hàm đổi một số nguyên hay số thực thành chuỗi ký tự (dạng số) và đưa vào biến s. Nếu là số nguyên i (dạng 1) thì bạn có thể chỉ thị thêm chiều dài của chuỗi n. Nếu là số thực r (dạng 2) thì ngoài việc chỉ thị chiều dài của chuỗi n, bạn có thể chỉ thị số chữ số thập phân tp.

169. Hàm SUCC

Cú pháp:

Succ(S);

Hàm trả về giá trị tiếp theo của S trong kiểu vô hướng đếm được.

170. Hàm SWAP

Cú pháp:

Swap(i);

Trong đó i có dạng Integer. Hàm Swap hoán chuyển vị trí Byte thấp và Byte cao trong i.

171. Hàm SWAPVECTORS (Dos Unit)

Cú pháp:

SwapVectors;

Hàm thay đổi giá trị hiện hành của bảng Vector ngắt bằng bảng đã được cất giữ trước khi bắt đầu thực hiện chương trình.

172. Thủ tục TEXTBACKGROUND (Crt Unit)

Cú pháp:

TextBackGround(<màu>);

Thay đổi màu nền bằng màu chỉ ra trong <mau>.

173. Thủ tục TEXTCOLOR (Crt Unit)

Cú pháp:

TextColor(<màu>);

Trong đó <màu> có dạng Word. Thủ tục dùng để thay đổi màu của ký tự bằng màu chỉ ra trong <màu>.

174. Hàm TEXTHEIGHT (Graph Unit)

Cú pháp:

TextHeight(s);

Trong đó s có dạng String. Kết quả của hàm có dạng Word là chiều cao (số Pixel) của s.

175. Thủ tục TEXTMODE (Crt Unit)

Cú pháp:

TextMode(mode);

Thủ tục này dùng để chọn Mode của văn bản.

176. Hàm TRUNC

Cú pháp:

Trunc(x);

Kết quả là phần nguyên của số thực x có dạng dữ liệu là số nguyên.

Ví dụ:

Trunc(12.76);

Kết quả:

12

177. Thủ tục TRUNCATE

Cú pháp:

Truncate(f);

Trong đó f là biến tệp. Thủ tục này cắt bỏ phần dữ liệu từ vị trí hiện thời của con trỏ trong tệp f.

178. Hàm UPCASE

Cú pháp:

Upcase(c);

Kết quả là giá trị hoa của c (dạng char) nếu c là chữ thường.

179. Thủ tục VAL

Cú pháp:

Val(s,r,code);

Thủ tục Val đổi chuỗi s thành số và đưa vào biến r. Nếu việc đổi thành công thì biến code (dạng Integer) được gán giá trị 0. Nếu việc đổi không thành công thì biến code được gán một giá trị là vị trí của kí tự trong biến s đã gây lỗi (ví dụ như kí tự không phải dạng số).

180. Hàm WHERE (Crt Unit)

Cú pháp:

Where;

Cho biết dòng đang chứa con trỏ trên màn hình.

181. Thủ tục WINDOW (Crt Unit)

Cú pháp:

Window(x1,y1,x2,y2);

Trong đó các giá trị x1, y1, x2, y2 có dạng Byte. Thủ tục giới hạn màn hình làm việc là hình chữ nhật có đỉnh trên bên trái là (x1, y1) và đỉnh dưới bên phải là (x2, y2).

Ví dụ:

Window(10,20,50,60);

182. Thủ tục WRITE và WRITELN

Cú pháp:

Write(<biểu thức>[,<tham số>]);

Writeln(<biểu thức>[,<tham số>]);

Thủ tục Write đưa các giá trị của <biểu thức> ra màn hình hoặc ra tệp. Writeln làm việc như Write nhưng sau khi đưa giá trị ra màn hình thì có mã kết thúc dòng. Hay nói cách khác, sau khi đưa giá trị của <biểu thức> ra màn hình hoặc máy in thì con trỏ nhảy xuống dòng dưới.

Ví dụ:

Write('Nguyễn Tuân Thúy');

Hoặc:

Writeln(''Nguyễn Tuân Thúy');

Trong trường hợp thứ hai, sau khi đưa kết quả ra màn hình, con trỏ tự động nhảy xuống dòng dưới.

Lệnh này còn có thể kết hợp với hàm GotoXY để định vị trí con trỏ tại một vị trí nào đó trên màn hình trước khi thực hiện lệnh Wirte hoặc Writeln.

Ví dụ:

GotoXY(100,50);

Writeln('Tuan Thúy');

Phụ lục 2

THÔNG BÁO LỖI TRONG TURBO PASCAL

1. CÁC LỖI KHI BIÊN DỊCH CHƯƠNG TRÌNH (COMPILER ERROR)

Mã	Thông báo lỗi	Nghĩa
1	Out of memory	Đầy bộ nhớ
2	Identifier Expected	Thiếu định danh (tên)
3	Unknown Identifier	Không hiểu định danh này
4	Duplicate Identifier	Định danh trùng nhau (2 lần)
5	Syntax Error	Sai cú pháp
6	Error in Real constant	Lỗi về hằng số thực
7	Error in Integer	Lỗi về hằng số nguyên
8	String constant exceed line	Hằng chuỗi vượt quá chiều dài quy định
9	Too many nested File	Quá nhiều tập tin lồng nhau
10	Unexpected end of file	Chưa kết thúc tập tin
11	Line too long	Dòng quá dài
12	Type identifier expected	Thiếu định danh kiểu
13	Too many open file	Mở quá nhiều tập tin
14	Invalid filename	Tên tập tin không hợp lệ
15	File not found	Không tìm thấy tập tin
16	Disk full	Đĩa đã đầy
17	Invalid compiler Directives	Chỉ thị dịch không đúng
18	Too many files	Quá nhiều tệp tin
19	Undefined type in pointer definition	Không định nghĩa kiểu cho con trỏ
20	Variable indentifier	Chưa định nghĩa biến
21	Error in Type	Lỗi về kiểu dữ liệu

22	Structure too large	Cấu trúc quá lớn
23	Set base type out of Range	Vượt phạm vi kiểu tập hợp
24	File components may not be files or objects	Thành phần tập tin không thể là các tập tin hay các đối tượng
25	Invalid string length	Chiều dài chuỗi không đúng
26	Type mismatch	Kiểu không phù hợp
27	Invalid subrange base type	Kiểu miền con không hợp lệ
28	Lower bound greater than Upper bound	Giới hạn dưới lớn hơn giới hạn trên
29	Ordinal Type expected	Thiếu kiểu thứ tự
30	Integer constant expected	Thiếu hằng số nguyên
31	Constant expected	Thiếu hằng số
32	Integer or real constant expected	Thiếu hằng số nguyên hay số thực
33	Type identifier expected	Thiếu định danh kiểu
34	Invalid functionresult Type	Kiểu kết quả của hàm không hợp lệ
35	Begin expected	Thiếu Begin
36	End expected	Thiếu End
37	Integer expression expected	Thiếu biểu thức số nguyên
38	Ordinal expression expected	Thiếu biểu thức thứ tự
39	Boolean expression expected	Thiếu biểu thức luận lý
40	Operand do not match operator	Toán hạng không phù hợp với toán tử
41	Error in expresstion	Biểu thứ sai
42	Illegal assigment	Lệnh gán không đúng
43	Field identifier expected	Thiếu định danh trường
44	Object file too large	Tập tin đối tượng quá lớn
45	Undefined External	Chưa định nghĩa chương trình ngoài
46	Invalid objéct-file record	Mẩu tin đối tượng không đúng
47	Code segment too large	Vùng code segment quá lớn
48	Data segment too large	Vùng data segment quá lớn
49	Do expected	Thiếu Do
50	Invalid public definition	Khai báo chung không hợp lệ
51	Invalid Extrn definition	Khai báo tham chiếu không hợp lệ

52	Too many Extrn definitons	Khái báo tham chiếu quá nhiều
53	OF expected	Thiếu OF
54	Interface expected	Thiếu Interface
55	Invalid relocatable reference	Tham chiếu đến đối tượng không đúng
56	THEN expected	Thiếu THEN
57	TO Or DOWNTO expected	Thiếu TO hay DOWTO
58	Undefined forward	Chưa khai báo trước
59	Too many procedure	Quá nhiều thủ tục
60	Invalid typecast	Chuyển đổi kiểu không hợp lệ
61	Division by zero	Chia cho số 0
62	Invalid file type	Kiểu tập tin không hợp lệ
63	Can not read or write variables of this type	Không thể đọc hoặc ghi dữ liệu kiểu này
64	Pointer variable expected	Thiếu biến con trỏ
65	String variable expected	Thiếu biến kiểu Chuỗi kí tự
66	String experession expected	Thiếu biểu thức Chuỗi kí tự
67	Circular unit reference	Tham chiếu các unit vòng tròn
68	Unit name mismatch	Tên unit không phù hợp với tập tin .TPU
69	Unit version mismatch	Các phiên bản của unit không phù hợp
70	Duplicate Unit name	Khai báo 2 lần tên unit
71	Unit file format error	Dạng tập tin TPU không đúng unit
72	Implementation expected	Thiếu Implementation
73	Constant and case tupe does not match	Các hằng và kiểu của case không khớp
74	Record variable expected	Thiếu biến record
75	Constant out of range	Hằng vượt quá phạm vi
76	File variable expected	Thiếu biến kiểu tập tin
77	Pointer expression expected	Thiếu biểu thức con trỏ
78	Integer and Real expression expected	Thiếu biểu thức con trỏ
79	Label not within current block	Nhãn không được ở ngoài khối hiện hành
80	Label already defined	Nhãn đã được định nghĩa

81	Undefined label in preceding statementpart	Nhãn chưa định nghĩa trong phần trước
82	invalid @ argument	Đối số của toán tử @ không hợp lệ
83	Unit expected	Thiếu unit
84	";" expected	Thiếu ;
85	":" expected	Thiếu :
86	"," expected	Thiếu ,
87	"(" expected	Thiếu (
88	")" expected	Thiếu)
89	"=" expected	Thiếu =
90	":=" expected	Thiếu :=
91	"[" or "(." expected	Thiếu [hoặc (.
92	"]" or ".)" expected	Thiếu] hoặc .)
93	"." expected	Thiếu .
94	".." expected	Thiếu ..
95	Too many variable	Quá nhiều biến
96	Invalid or control variable	Biến điều khiển không đúng
97	Integer variable expected	Thiếu biến integer
98	Files are not allowed here	Không được phép sử dụng các tập tin ở đây
99	String length mismatch	Chiều dài chuỗi không đúng
100	Invalid ordering of fields	Thứ tự của các trường không hợp lệ
101	String constant expected	Thiếu hằng kiểu chuỗi
102	Integer or Real variable expected	Thiếu biến kiểu integer hay real
103	Ordinal variable expected	Thiếu biến thứ tự
104	Inline error	Lệnh Inline sai
105	Character expression expected	Thiếu biểu thức kí tự
106	Too many relocation items	Quá nhiều mục chuyển vị
107	Case constantout of range	Các hằng trong lệnh case vượt phạm vi
108	Error in statement	Sai trong câu lệnh
109	Cannot call an interrupt procedure	Không thể gọi các thủ tục ngắt
110	Must be in 8087 mode to compile this	Phải biên dịch ở chế độ 8087
111	Target address not found	Không tìm thấy dịa chỉ đích

112	Include files are not allowed here	Các tập tin include không được phép ở đây
113	Nil expected	Thiếu Nil
114	Invalid qualifier	Định tính không hợp lệ
115	Invalid variable reference	Tham chiếu biến không đúng
116	Too many symbols	Quá nhiều kí hiệu
117	Statement part too long	Phần lệnh quá dài (vượt quá 24KB)
118	Files must be Var parameters	Tập tin phải là tham biến
119	Too many conditionnal symbols	Quá nhiều kí tự điều kiện
120	Misplaced conditional directive	Chỉ thị điều kiện đặt sai vị trí
121	ENDIF directive missing	Thiếu chỉ thị ENDIF
122	Error in initial conditional defines	Lỗi khi định nghĩa điều kiện
123	Header does not match previous definition	Tiêu đề không phù hợp với định nghĩa
124	Critical disk error	Lỗi đĩa
125	Cannot evaluate this expression	Không thể đánh giá biểu thức này
126	Expression incorectly terminated	Chấm dứt biểu thức không đúng
127	Invalid format specifier	Định khuôn dạng không hợp lệ
128	Invalid indirect reference	Tham chiếu gián tiếp không hợp lệ
129	Structured variables are not allowed here	Không được sử dụng biến cấu trúc ở đây
130	Cannot evaluate without SYSTEM unit	Không thể xác định vì không có unit SYSTEM
131	Cannot acces this symbol	Không thể truy xuất đến kí hiệu này
132	Invalid floating-point operation	Dấu chấm động không hợp lệ
133	Cannot compile overlays to memory	Không thể biên dịch phủ lấp vào bộ nhớ
134	Procedure Or Function variable expected	Thiếu biến thủ tục hay hàm
135	Invalid procedure or function reference	Tham chiếu thủ tục hay hàm không hợp lệ
136	Cannot overlay this unit	Không thể phủ lấp unit này
137	Object type expected	Thiếu kiểu object

138	Local object types are not allowed	Không chấp nhận kiểu object cục bộ
139	Virtual expected	Thiếu virtual
140	Method identifier expected	Thiếu định danh phương pháp
141	Virtual constructors are not allowed	Constructor ảo không được phép
142	Constructor identifier expected	Thiếu định danh constructor
143	Destructor identifier expected	Thiếu định danh destructor
144	Fail only allowed within constructors	Chỉ cho phép trong constructor
145	Invalid combination of operator and operance	Kết hợp toán hạng và toán tử không hợp lệ
146	Memory reference expected	Thiếu tham chiếu bộ nhớ
147	Cannot add or subtract relocatable symbols	Không thể thêm bớt kí tự định vị
148	Invalid register combination	Kết hợp thanh ghi không hợp lệ
149	286/287 instruction are not enabled	Các chỉ thị 286/287 không được dùng
150	Invalid symbol reference	Tham chiếu kí hiệu không đúng
151	Code generation error	Lỗi phần code (mã)

2. CÁC LỖI KHI THỰC HIỆN CHƯƠNG TRÌNH (RUN-TIME ERROR)

Lỗi Run-time 3 xẩy ra khi chương trình đang được thực hiện. Khi xẩy ra lỗi, Turbo Pascal hiển thị thông báo lỗi như sau:

Run-time error NNN at XXXX : YYYY

Trong đó NNN là mã lỗi, còn XXXX là segment và YYYY là offset, tức địa chỉ xảy ra lỗi.

Dưới đây phân loại thành các nhóm như sau:

Lỗi liên quan đến hệ điều hành

2	File not found	Không tìm thấy tập tin
3	Path not found	Không tìm thấy đường dẫn
4	Too many open files	Mở quá nhiều tập tin
5	File access denied	Không truy cập được tập tin
6	Invalid file handle	Quản lí tập tin không hợp lệ
12	Invalid file acces code	Mã truy cập tập tin không hợp lệ

15	Invalid drive number	Số hiệu ổ đĩa không đúng
16	Cannot remove current directory	Không thể xóa thư mục hiện hành
17	Cannot rename across drives	Không thể đổi tên tập tin khác ổ đĩa

Lỗi do nhập xuất (I/O Error)

100	Disk read error	Lỗi khi đọc đĩa
101	Disk write error	Lỗi khi ghi vào đĩa
102	File not assigned	Chưa gán tên tập tin
103	File not open	Chưa mở tập tin
104	File not open for input	Chưa mở tập tin để ghi
105	File not open for output	Chưa mở tập tin để đọc
106	Invalid numeric format	Khuôn dạng số không hợp lệ

Lỗi vật lí (Critical error)

150	Disk is write-protected	Đĩa đã dán bảo vệ chống ghi
151	Unknown unit	Không hiểu unit
152	Drive not ready	Ổ đĩa chưa sẵn sàng (chưa đóng cửa)
153	Unknown command	Không hiểu lệnh
154	CRC (Cyclic Redundanas Check) error in data	Lỗi ở mã kiểm tra dữ liệu
155	Bad drive request structure length	Khoảng yêu cầu trên đĩa bị hư
156	Disk seek error	Lỗi đĩa khi tìm dữ liệu
157	Unknown media type	Không hiểu kiểu của dữ liệu
158	Sector not found	Không tìm thấy cung từ
159	Printer out of paper	Máy in không có giấy
160	Device write fault	Thiết bị ghi bị hư
161	Device read fault	Thiết bị đọc bị hư
162	Harware failure	Phần cứng bị hư
200	Division by zero	Chia cho số 0
201	Range check error	Vượt quá phạm vi
202	Stack overflow error	Tràn vùng Stack
203	Heap overflow error	Tràn vùng Heap
204	Invalid pointer operation	Con trỏ không hợp lệ

205	Fôating-point overflow	Tràn dấu chấm động
206	Floating-point underflow	Không đủ cho dấu chấm động
207	Invalid floating-point operation	Dấu chấm động không hợp lệ
208	Overlay manager not installed	Chưa thiết lập chế độ quản lí phủ lấp
209	Overlay file read error	Lỗi đọc tập tin phủ lấp
210	Object not initialized	Chưa khởi tạo object
211	Call to abstract method	Gọi tới phương pháp abstract
212	Stream registration error	Đăng ký stream sai
213	Collection index out of range	Chỉ mục vượt quá phạm vi
214	Collection overflow error	Lỗi tràn collection

Phụ lục 3

PHẦN MỀM THIẾT KẾ ĐƯỜNG NOVA

Phiên bản 3.5 - 91.7 MB. Dùng với AutoCAD 14

NOVA-TDN là sản phẩm hợp tác giữa Công ty Hài Hoà và Hãng VIA NOVA Na Uy.

- *Các Tác giả:* KS Hồ Việt Hải, KS Nguyễn Hải Phương, KS Phan Mạnh Dân, KS Nguyễn Thạch Hưng, KS Trần Trọng Nghĩa, KS Phạm Văn Trung.

- *Cố vấn chuyên môn:* PGS-PTS Nguyễn Viết Trung, PTS Nguyễn Nhật Quang, Th.S Nguyễn Bá Sinh, KS Chư Quang Định, Th.S Võ Thanh Tùng, Th.S Nguyễn Đức Hát.

- *Địa chỉ liên hệ:* + Hà Nội : 123 Trúc Bạch, Quận Ba Đình – *Tel 04.8290413*

+ TP Hồ Chí Minh: 29 Bùi Thị Xuân, Quận 1–*Tel 08.9252227.*

Sau đây giới thiệu sơ lược về cách sử dụng để bước đầu làm quen. Chi tiết xem thêm trong NOVA Help.

I. KHÁI QUÁT CHUNG

Phần mềm NOVA có thể giúp kỹ sư đường lên các bản vẽ bình đồ, cắt dọc, cắt ngang. Thiết kế đường đỏ trên cắt dọc. Tính diện tích đào đắp trên cắt ngang thiết kế, tính khối lượng đào đắp nền đường.

Người kỹ sư thiết kế có thể tạo mẫu bản vẽ cắt dọc, cắt ngang theo các đề mục mong muốn cả về tên gọi và cự ly các dòng đề mục.

1. Vào và ra khỏi NOVA

Khi vào Nova, đơn giản nhất là nhấn vào Icon NOVA trên màn hình *(Destop)*, ra trang File mở đầu *(New File)* như Hình N1. Trang này gồm các *menu* và thanh công cụ như AutoCAD, có thêm một số *menu* tiếng Việt như **Địa hình, Bình đồ, TD-TN** (trắc dọc- trắc ngang), **Phụ trợ**.

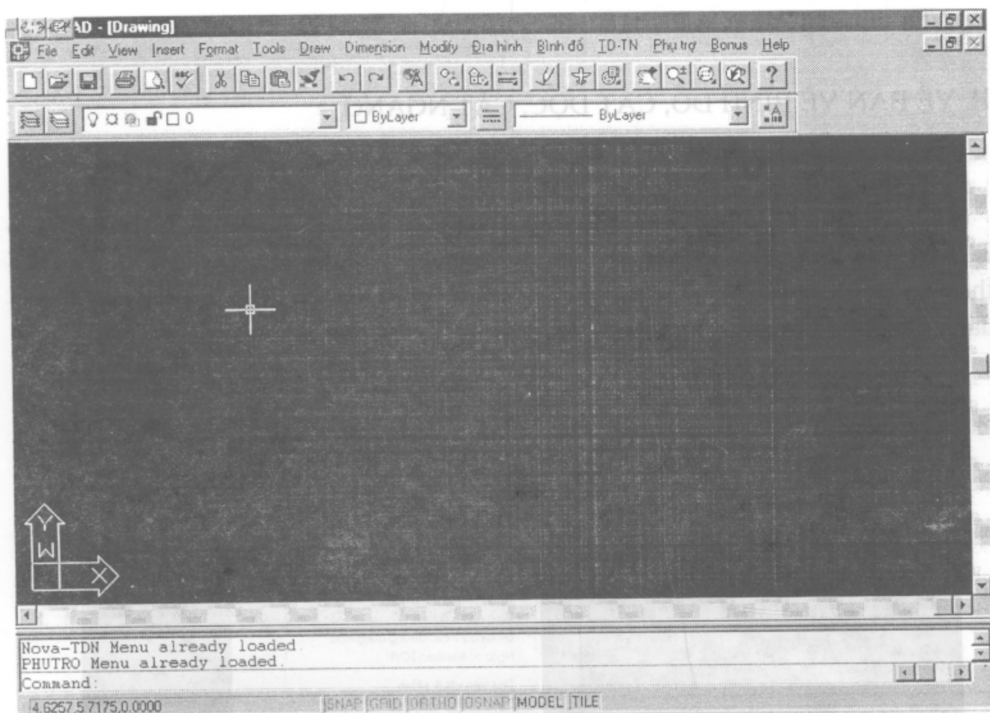
Mỗi menu trên cho một *bảng chọn* với nhiều tình huống lập bản vẽ. Đơn giản nhất là:

- **Địa hình** dùng để nhập số liệu đo đạc ngoài hiện trường theo Tiêu chuẩn Việt Nam.

- **Bình đồ** để vẽ bình đồ tự nhiên và thiết kế bình đồ.

- **TD-TN** để vẽ cắt dọc, cắt ngang tự nhiên và thiết kế cắt dọc, cắt ngang. Tính khối lượng.

Thoát khỏi NOVA: Vào *File → Exit*.



Hình N1: New File khi vào NOVA

2. Ghi tệp (Save File)

- *Ghi tệp dữ liệu:* Nói chung sau khi nhập số liệu xong phần cơ bản (như số liệu vẽ cắt dọc) nên Save File ngay bằng cách nhấn vào mục *Ghi tệp* trên hình N3. Đặt tên File (thí dụ *HMuc-CN*) theo đường dẫn D:\ D-Hoa\ Ve-NOVA\ HMuc-CN. Tên File này được lưu giữ trong NOVA và trong Explorer nhưng chỉ mở lại được trong NOVA.

- *Ghi tệp bản vẽ:* Nhập số liệu xong, lên các bản vẽ, tiếp tục Save File bản vẽ trong AutoCAD (thí dụ tên File *C-Doc-HMuc*). File này có thể mở được trong NOVA và trong Explorer.

3. Mở lại File đã có

- Mở lại File số liệu (như HMuc-CN) phải vào NOVA → Vào menu *Địa hình* → nhấn *Nhập số liệu theo TCVN* → nhấn *Mở tệp* → tìm tên File theo đường dẫn → nhấn đúp chuột trái lên tên File sẽ mở được.

Cách khác: gõ **nsl** sau Command → Enter → ra Nhập dữ liệu → nhấn Mở tệp ..

- Mở lại File bản vẽ (như C-Doc-HMuc) : mở trong NOVA và trong Explorer đều được.

4. In bản vẽ

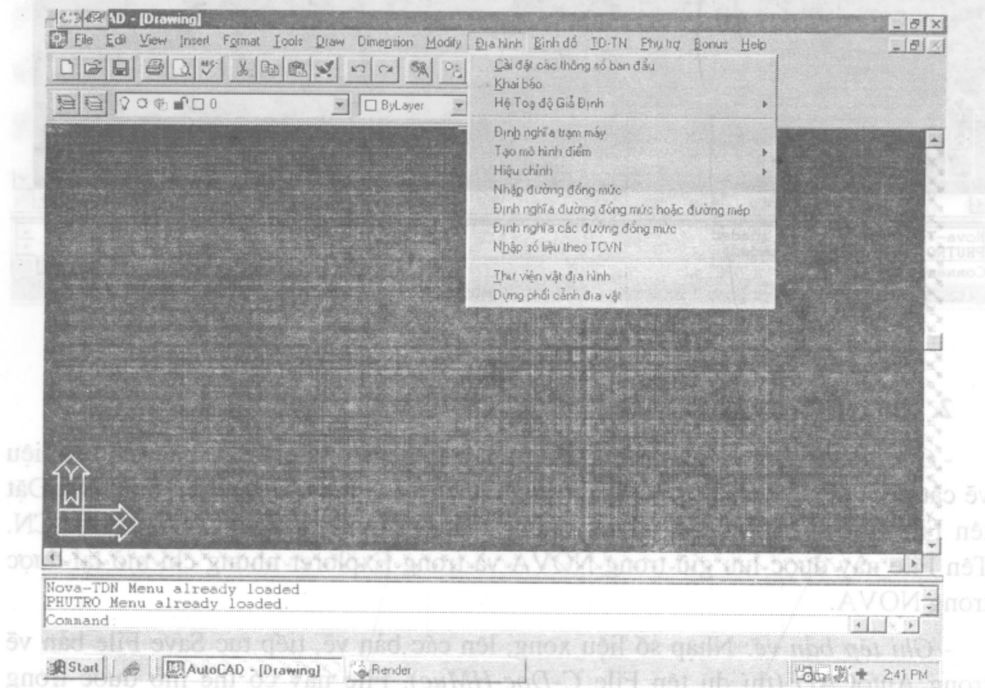
Như in trong AutoCAD

II. VẼ BẢN VẼ BÌNH ĐỒ, CẮT DỌC, CẮT NGANG

A. Bản vẽ hiện trạng

1. Vào Menu *Địa hình* → ra Bảng chọn như Hình N2 → *Nhập số liệu theo TCVN* → ra hộp thoại *Nhập dữ liệu* như hình N3. Nhập số liệu vẽ cắt dọc có thể theo khoảng cách lẻ (hoặc cộng dồn), cao độ tự nhiên. Nếu dùng khoảng cách lẻ đánh dấu mục *Cọc theo KC lẻ* trên Hình N3.

Cách khác: Vào NOVA, *Command* : **nsl** → *Enter* → cũng ra Hình N3.



Hình N2: Bảng chọn menu Địa hình

2. Nhập xong nhấn *Ghi tệp* → ra hộp thoại *Save as* → đặt tên File vào mục *File name* (thí dụ Hmuc-CN), tạo đường dẫn trong *Save in* (thí dụ D:\ D-Hoa\ Ve-NOVA\ Hmuc-CN). Tên File có đuôi *.ntd. Nhấn *Save* → trở về *Nhập dữ liệu* → nhấn *Nhập TN* (trắc ngang) → ra hộp thoại như Hình N4.

Nhập dữ liệu Nova-TDN - D:\D-Hoa-D\Ve-NOVA\HMuc-CN.ntd

Mô tả:

T.T	Tên cọc	KCách	CĐTN	CĐTK	Góc HT	B.Kính	C.dồn
1	1	0.0000	3.1500		180:0:0		
2	2	10.0000	3.1400		180:0:0		
3	3	20.8400	4.6400		180:0:0		
4	4	31.6800	5.9200		180:0:0		
5	5	41.0500	6.6000		180:0:0		
6	6	46.0500	5.9200		180:0:0		
7	M3'	62.0500	5.8400		180:0:0		
8	7	78.0500	5.8400		180:0:0		
9	8	83.0500	5.9100		180:0:0		
10	9	92.4200	5.7200		180:0:0		
11	10	03.2600	4.2100		180:0:0		

Cọc theo KC lẻ Mia theo KC lẻ CĐ Mia tương đối

Thao tác tệp: Tệp mới | Ghi tệp.. | Mở tệp..

ĐC chênh cao | Nhập TN<<

Nhận | Hủy | Tính D/2 | Tính C.dồn

Hình N3: *Hộp thoại Nhập dữ liệu cắt dọc theo khoảng cách cộng dồn và cao độ tự nhiên*

AutoCAD - [Drawing]

File Edit View Insert Format Tools Draw Dimension Modify Địa hình Bình đồ TD-TN Phụ trợ Bonus Help

Nhập dữ liệu Nova-TDN - D:\D-Hoa-D\Ve-NOVA\HMuc-CN.ntd

Mô tả:

T.T	Tên cọc	KCách	CĐTN	CĐTK	Góc HT	B.Kính	C.dồn
1	1	0.0000	3.1500		180:0:0		
2	2	10.0000	3.1400		180:0:0		
3	3	20.8400	4.6400		180:0:0		
4	4	31.6800	5.9200		180:0:0		
5	5	41.0500	6.6000		180:0:0		
6	6	46.0500	5.9200		180:0:0		
7	M3'	62.0500	5.8400		180:0:0		
8	7	78.0500	5.8400		180:0:0		
9	8	83.0500	5.9100		180:0:0		
10	9	92.4200	5.7200		180:0:0		
11	10	03.2600	4.2100		180:0:0		

Nhập Mia cho cọc 1

Các điểm Mia trái

T.T	KCách	CaoĐộ	SHV	GhiChú
1	5.7000	3.1500	0	
2	6.7000	3.1500	0	
3	8.0000	7.0200	0	
4	10.0000	7.0200	0	
5	13.0000	6.3600	0	
6	15.8900	5.7200	0	
7				
8				

Các điểm Mia phải

T.T	KCách	CaoĐộ	SHV	GhiChú
1	5.7000	3.1500	0	
2	6.7000	3.1500	0	
3	8.0000	7.0600	0	
4	10.0000	7.0600	0	
5	13.0000	7.2700	0	
6	24.0100	8.0400	0	
7				
8				

Cọc theo KC lẻ Mia theo KC lẻ CĐ Mia tương đối

Thao tác tệp: Tệp mới | Ghi tệp.. | Mở tệp..

ĐC chênh cao | Nhập TN>>

Nhận | Hủy | Tính D/2 | Tính C.dồn

Minh họa Trắc Dọc

Command:
Command:

Start | AutoCAD - [Drawing] | Render | 2:46 PM

Hình N4: *Nhập số liệu cắt ngang tự nhiên*

3. Nhập số liệu cắt ngang theo khoảng cách ngang cộng dồn tính từ tim. Nhập số liệu bên trái trước, sau đến bên phải.

Nhập xong tiếp tục Save vào tên File HMuc-CN.

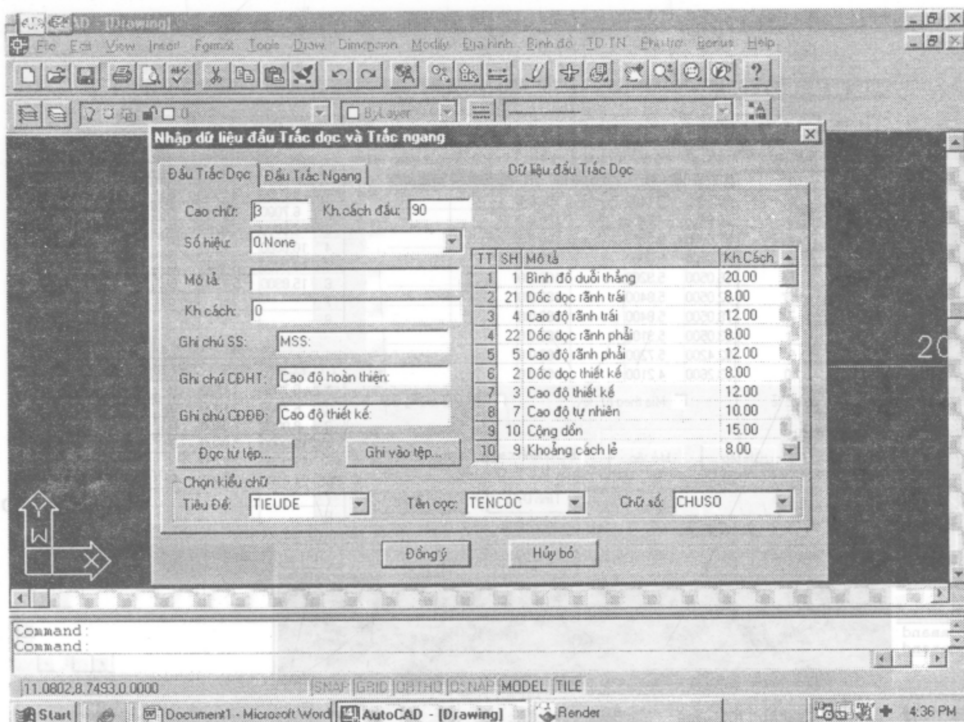
4. Vào menu *Bình đồ* → *Vẽ tuyến theo TCVN* → ra hộp thoại *Read Data from* , tìm tên File nhập dữ liệu theo đường dẫn **D:\ D-Hoa\ HMuc-CN** → mở File ra hộp thoại *Nhập gốc tuyến* → nhấn mục *Điểm đầu* → nhấn điểm chọn trên màn hình → hiện ra hình vẽ Bình đồ tuyến.

Cách khác: gõ chữ **t** sau *Command* rồi *Enter* cũng ra ngay hộp thoại *Read Data from* , tiếp tục làm như trên.

5. Vào menu TD-TN → *Trắc dọc tự nhiên* → nhấn *Vẽ trắc dọc tự nhiên* → ra hộp thoại *Trắc dọc tự nhiên của tuyến đường thứ 1.* Nhập tỷ lệ cắt dọc ngang 1:1000, đứng 1:100.

Nhấn *Khai mẫu bảng* ra hộp thoại *Nhập dữ liệu đầu trắc dọc và trắc ngang* như Hình N5. Vào mục *Đầu trắc dọc.* Khai xong mẫu cắt dọc theo ý muốn, nhấn *Đồng ý* → OK ra bản vẽ cắt dọc.

Cách khác: *Command* : **td** → *Enter* cũng ra hộp thoại *Trắc dọc tự nhiên của tuyến đường thứ 1.* Tiếp tục làm như trên.



***Hình N5:** Hộp thoại Nhập dữ liệu đầu trắc dọc và trắc ngang*

Lưu ý:

- NOVA chỉ vẽ cắt dọc sau khi vẽ bình đồ.

- Hộp thoại Hình N5 có mục *Số hiệu, Mô tả* để ta chuyển đổi tên hàng trong bảng theo cách gọi *của ta* thành cách gọi *của máy.*

Thí dụ: Ta gọi là *Bình đồ duỗi thẳng*, nhưng tương ứng *Số hiệu 1* trong máy gọi là *Bình đồ sơ lược*. Đây là sự chuyển đổi để máy thực hiện đúng các lệnh sẽ thực hiện sau này khi lên bản vẽ (như tính độ dốc, cao độ thiết kế v.v...).

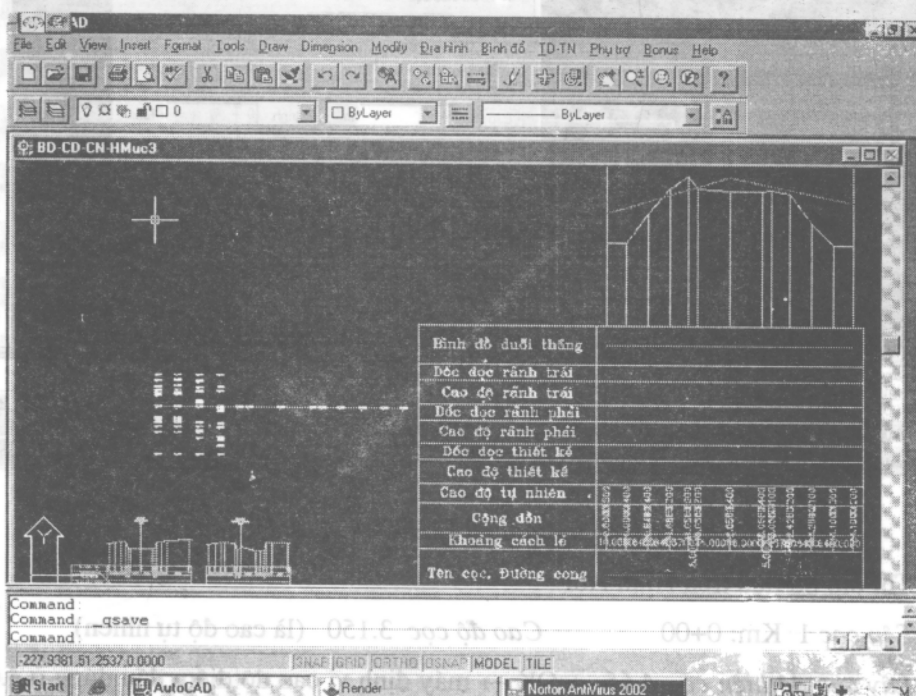
6. Vào menu TD-TN → *Trắc ngang tự nhiên* → nhấn *Vẽ trắc ngang tự nhiên* → ra hộp thoại *Trắc ngang tự nhiên của tuyến đường thứ 1*. Nhập tỷ lệ vẽ cắt ngang 1:200.

Nhấn *Khai mẫu bảng* ra hộp thoại *Nhập dữ liệu đầu trắc dọc và trắc ngang* như Hình N5. Vào mục *Đầu trắc ngang*. Khai xong mẫu cắt ngang theo ý muốn, nhấn *Đồng ý* → OK ra bản vẽ cắt ngang.

Cách khác: *Command* : **tn** → *Enter* cũng ra hộp thoại *Trắc ngang tự nhiên của tuyến đường thứ 1*. Tiếp tục làm như trên.

7. Vẽ xong bình đồ, cắt dọc, cắt ngang tự nhiên nên Save vào một File bản vẽ AutoCAD (thí dụ BD-CD-CN- HMuc). Sau đó bắt đầu thiết kế đường đỏ.

Kết quả vẽ xong bình đồ, cắt dọc, cắt ngang được thể hiện trên màn hình như Hình N6. (Hình này thể hiện thí dụ mới nhập số liệu có 4 cắt ngang)



Hình N6: *Bản vẽ Bình đồ, cắt dọc, cắt ngang tự nhiên*

B. Thiết kế đường đỏ trên cắt dọc.

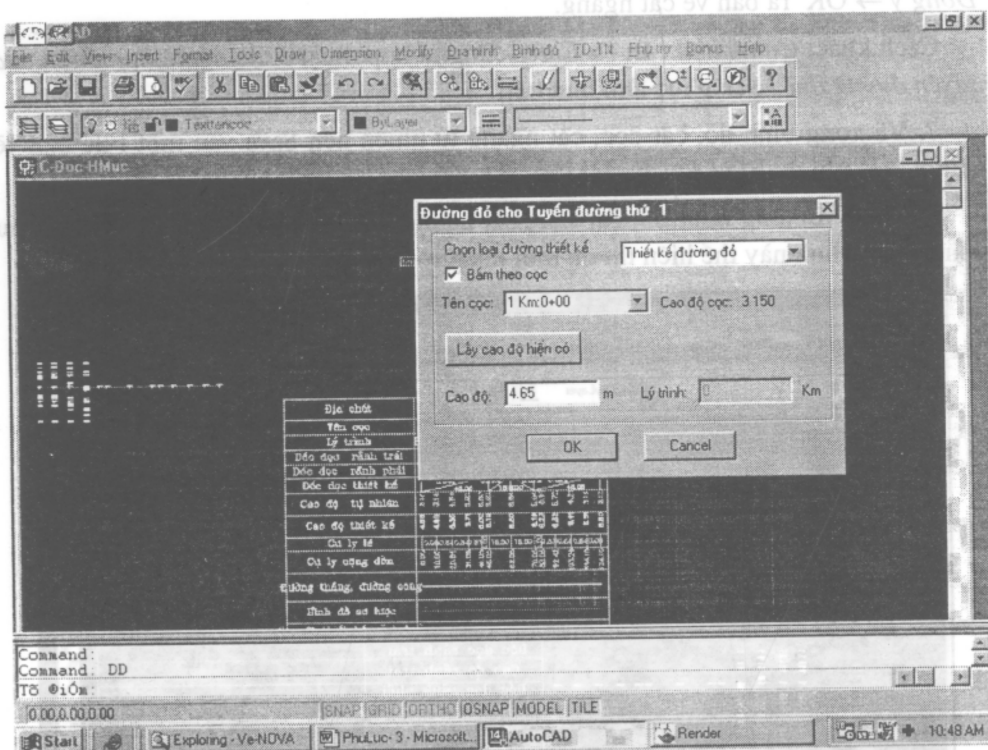
Thiết kế đường đỏ trên cắt dọc có thể theo 2 cách:

- Xác định cao độ khống chế các điểm đổi dốc trên cắt dọc rồi nối lại, máy sẽ tính tiếp ra số liệu độ dốc, cao độ thiết kế, cao độ thi công.

- Từ một cao độ ban đầu, chọn độ dốc thiết kế theo ý muốn đến điểm khống chế thứ hai.

1. Vào menu TD-TN → nhấn *Thiết kế trắc dọc* → nhấn một điểm trên màn hình ra hộp thoại *Đường đỏ cho tuyến đường thứ 1* như Hình N7. Đánh dấu mục *Bấm theo cọc*.

Nếu thiết kế theo cao độ, nhập trị số cao độ thiết kế theo cọc đổi dốc → OK

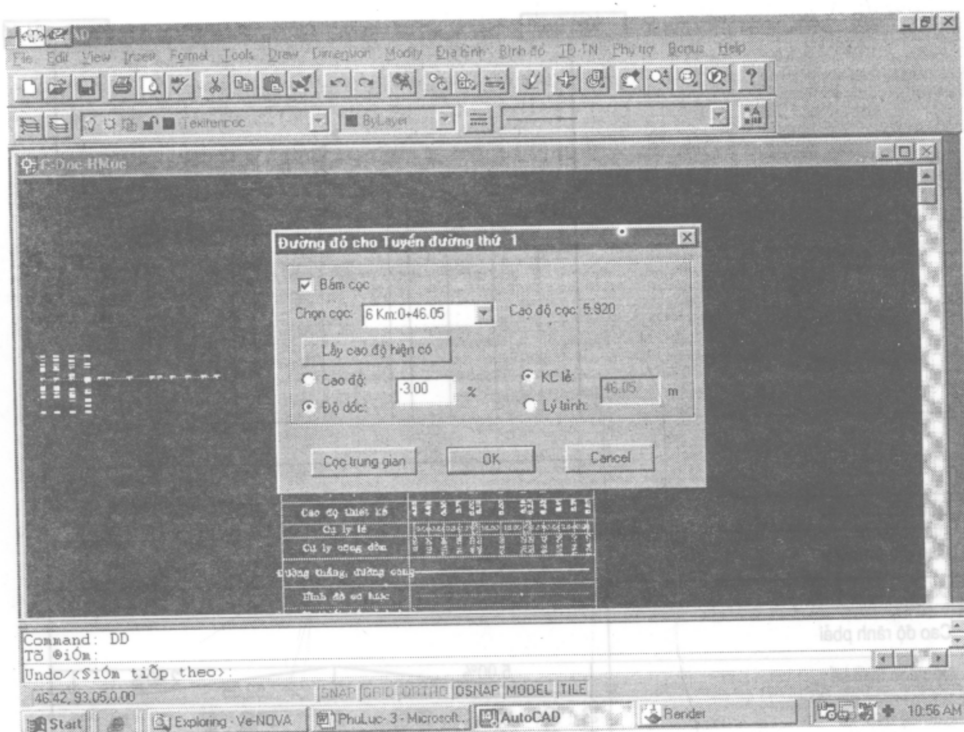


Hình N7: *Hộp thoại thiết kế đường đỏ điểm đầu tiên*

Thí dụ: Số liệu trên Hình N6:

Tên cọc 1 Km: 0+00 *Cao độ cọc* 3.150 (là cao độ tự nhiên)

Lấy cao độ hiện có 4.65 → OK → máy định vị cao độ điểm đầu đúng ở 4.65, đồng thời ra Hình N8.



Hình N8: Thiết kế đường đỏ điểm tiếp theo

Di chuột đến điểm tiếp theo, làm tương tự.

Nếu thiết kế theo độ dốc, trên hộp thoại Hình N8 đánh dấu mục *Độ dốc*. Nhập độ dốc thiết kế (thí dụ –3%), chọn điểm cuối dốc → OK.

Quy định: dấu (-) là lên dốc, (+) là xuống dốc.

Đường đỏ thiết kế xong thể hiện như Hình N6.

2. Thiết kế xong, vào TD-TN chọn mục *Điền thiết kế*, máy sẽ điền các số liệu cao độ thiết kế, cao độ thi công, độ dốc/ chiều dài dốc từng đoạn v.v...

Cách khác: Command : **dtk** → Enter → ra hộp thoại *Điền thiết kế trắc dọc*. Dùng chuột chỉ vào đường đỏ. Đánh dấu vào các mục:

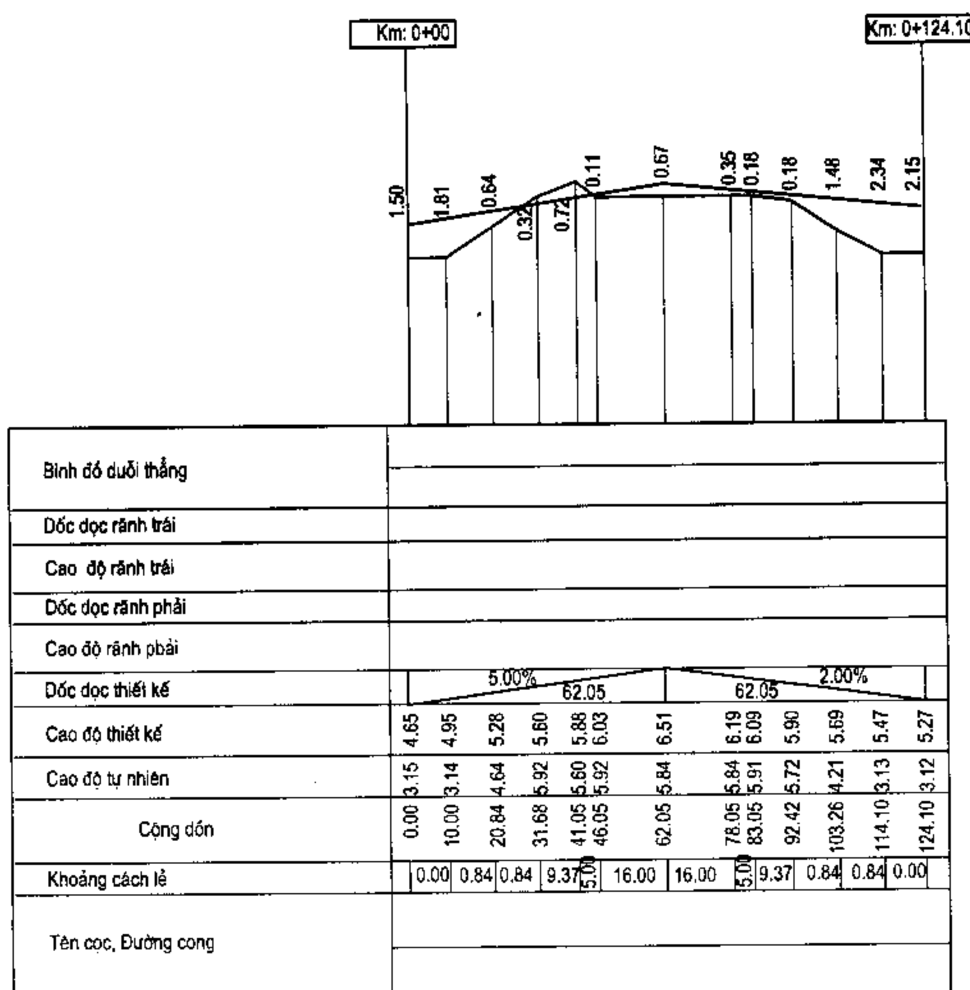
- Điền giá trị chênh cao (tức cao độ thi công)
- Dốc dọc tính đến đỉnh giao nhau

OK

Kết quả cho Hình N9.

C. Thiết kế cắt ngang

1. Vào TD-TN → *Thiết kế trắc ngang* → ra hộp thoại *Áp thiết kế trắc ngang* → nhấn mục *Sửa* ra hộp thoại *Nhập số liệu trắc ngang thiết kế* như Hình N10.



Hình N9: *Cắt dọc đã thiết kế xong.*

Số liệu nhập trên Hình N10 là thí dụ cho đường đô thị, có bó vỉa cao 0.15m nên nhập số liệu ở hai mục:

Cao lề : 0.15

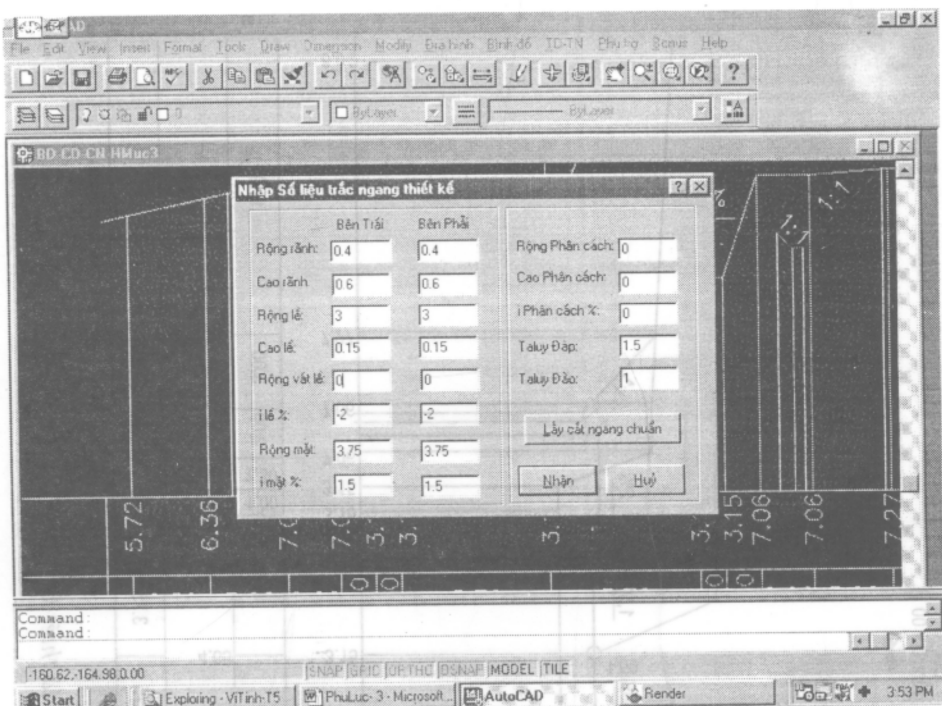
Rộng vát lề : 0.00

i lề % : -2

i là độ dốc ngang mặt đường, lề (hè) đường, quy ước:

- Dấu (+) là dốc từ tim đường ra mép đường.
- Dấu (-) là dốc từ mép đường vào tim đường.

Ở trên i lề tức là dốc ngang hè đường đô thị, thường dốc về phía tim đường (bó vỉa) nên có dấu (-), khác với lề đường ngoài đô thị thường dốc ra ngoài (dấu +).



Hình N10: *Nhập số liệu trắc ngang thiết kế*

Nhập số liệu xong nhấn *Nhận* , trở về hộp thoại *Áp thiết kế trắc ngang*, nhấn *Áp thiết kế*, ra kết quả các hình cắt ngang đã thiết kế như Hình N6. Chi tiết một cắt ngang xem Hình N11

2. Vào TD-TN → *Thiết kế trắc ngang* → *Điền thiết kế trắc ngang* → ra hộp thoại *Tuyến đường thứ 1* → đánh dấu vào các mục:

+ Điền độ dốc

+ Điền cao độ đường đỏ

+ Điền cao độ hoàn thiện

OK → ra cắt ngang thiết kế như Hình N11.

Lưu ý: Khi thiết kế đường có dải phân cách ở giữa, cao độ thiết kế vẫn là *cao độ mặt đường*, tức là *nằm trên đường ngang nối chân bó vỉa 2 bên dải phân cách.* Tuy nhiên, máy vẫn điền cao độ thiết kế ở vị trí tim trên đỉnh bó vỉa, người thiết kế nên chỉnh lại.

Đây chỉ là chi tiết nhỏ, nhưng khi một ngã tư giao nhau do 2 đơn vị thiết kế khác nhau, tại điểm giao nhau tim 2 đường giao nhau, được cơ quan quy hoạch cho trước cao độ mặt đường khống chế (Thí dụ 4.65 m), một tuyến đường có dải phân cách, một tuyến không có, lúc khớp nối cao độ mặt đường khống chế có thể bị lệch nhau bởi cao độ bó vỉa dải phân cách, (thí dụ dải phân cách cao 30 cm thì cao độ tim đường có dải phân cách sẽ là 4,95 m).

Coc 1
KmL 0 + 00

Đắp nền: 21,54m^2
Đào nền: 18,88m

Cao độ hoàn thiện: 1,65
Cao độ thiết kế: 4,65

1:1 1:1 2% 1,50% 1.50 1,50% 2% 1:1 1:1

Cao độ thiên nhiên: 5.72 6.36 7.02 7.02 3.15 3.15 3.15 3.15 3.15 7.06 7.06 7.27 8.04

Khoảng cách mia: 2.89 3.00 2.00 1.30 1.00 5.70 5.70 1.00 1.30 2.00 3.00 11.01

Cao độ thiết kế: 6.56 4.24 4.24 4.84 4.74 4.59 4.65 4.59 4.74 4.84 4.24 4.24 7.25

Khoảng cách lề thiết kế: 2.32 0.40 0.60 5 3.75 3.75 5 0.60 0.40 3.01

Hình N11: *Cắt ngang đã thiết kế*

D. Tính diện tích cắt ngang – Khối lượng nền đường

1. Vào TD-TN → Diện tích → Tính Diện tích → ra hộp thoại Tính diện tích của tuyến đường thứ 1 . Đánh dấu mục Đường tự nhiên lấy theo cắt ngang → OK

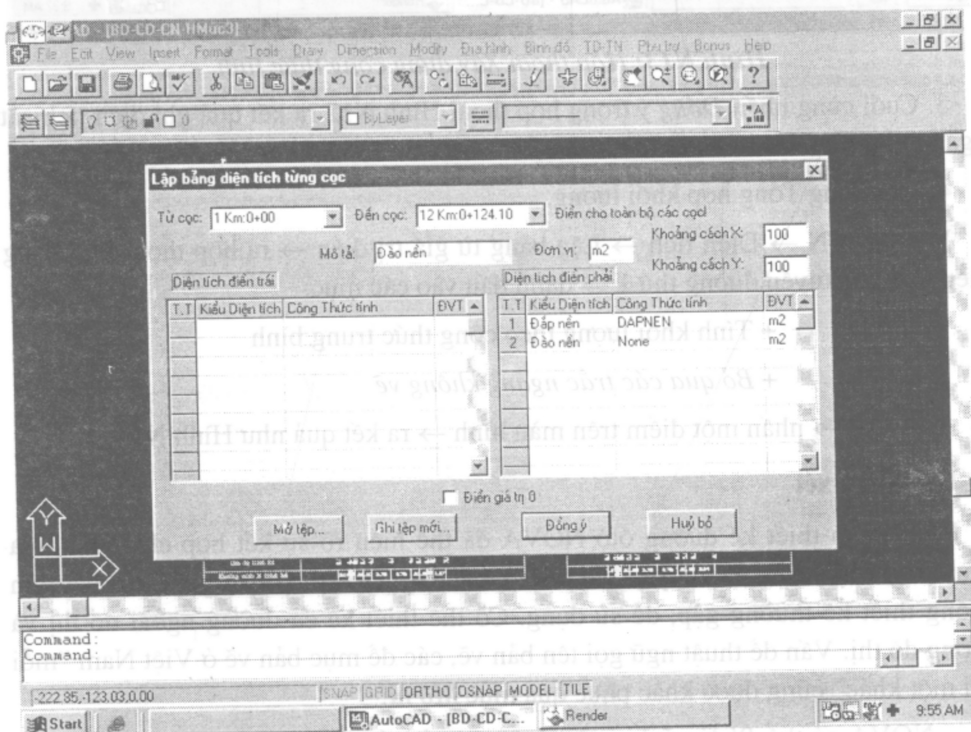
2. Vào TD-TN → *Diện tích* → *Điền giá trị Diện tích* → ra hộp thoại *Lập bảng diện tích từng cọc* (Hình N12) → Nhấn dòng 1 mục *Diện tích điền phải* (nếu muốn ghi trị số phía phải cắt ngang) → gõ chữ *Đắp nền* mục *Mô tả* → nhấn chữ *None* cột *Công thức tính* ra hộp thoại *Xây dựng công thức tính* (Hình N13) → nhấn ▾ mục *Mô tả* ra bảng chọn → nhấn *Đắp nền- DAPNEN* → OK → về *Lập bảng diện tích từng cọc* → nhấn *None* ra kết quả

Kiểu Diện tích	*Công Thức tính*
Đắp nền	DAPNEN

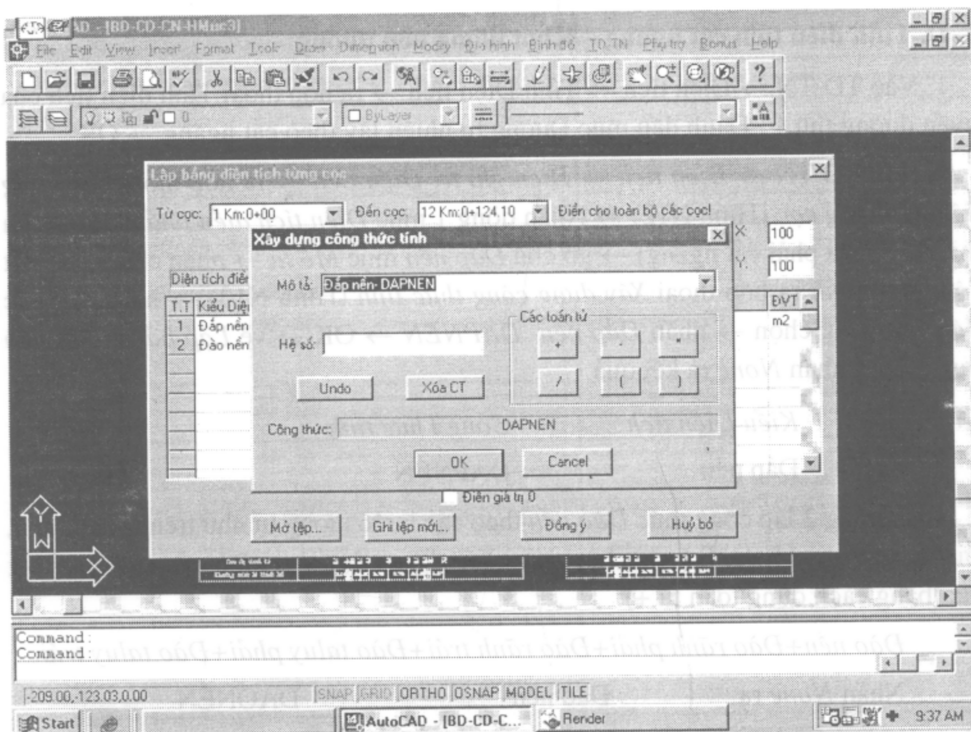
Nhấn dòng 2 lập công thức *Đào nền* theo các bước tương tự như trên. Tuy nhiên, với đào nền như Hình N11, qua hộp thoại *Xây dựng công thức tính* cần xác lập công thức(bằng cách dùng toán tử +):

Đào nền+Đào rãnh phải+Đào rãnh trái+Đào taluy phải+Đào taluy trái.

Nhấn *None* ra Đào nền DAONEN



Hình N12: Hộp thoại Lập bảng diện tích từng cọc



Hình N13: Hộp thoại Xây dựng công thức tính

3. Cuối cùng nhấn *Đồng ý* trong hộp thoại Hình N12 ra kết quả ghi diện tích cắt ngang như hình N11.

4. Lập Bảng Tổng hợp khối lượng.

Vào TD-TN → Diện tích → Lập bảng từ giá trị điền → ra hộp thoại Lập Bảng diện tích của tuyến đường thứ 1 → đánh dấu vào các mục:

+ Tính khối lượng theo công thức trung bình

+ *Bỏ qua các trắc ngang không vẽ*

OK → nhấn một điểm trên màn hình → ra kết quả như Hình N14.

III. Nhận xét

Phần mềm thiết kế đường ôtô NOVA đã thể hiện rõ sự kết hợp chặt chẽ giữa chuyên gia lập trình và chuyên gia thiết kế đường. Do đó đã bao gồm hầu hết tình huống thiết kế thường gặp, dễ sử dụng. Có thể thiết kế cả đường ngoài đô thị và đường đô thị. Vấn đề thuật ngữ gọi tên bản vẽ, các đề mục bản vẽ ở Việt Nam "mỗi nơi một khác" cũng được khắc phục qua mục *Mô tả*.

NOVA có thể *Phối cảnh tuyến đường thiết kế, Hoạt cảnh theo hành trình*, từ đó giúp ta thiết kế đường theo yêu cầu tầm nhìn xe chạy êm thuận, mỹ quan.

Từ cọc: 1Km:0 + 00
Tới cọc: 4Km:0 + 31,68

Tên cọc	Khoảng cách lẻ	Diện tích		Diện tích trung bình		Khối lượng	
		Đắp nền	Đào nền	Đắp nền	Đào nền	Đắp nền	Đào nền
1		21,54	18,28				
	10,00			23.29	16,22	232,90	162,20
2		25,03	14,15				
	10,84			22,93	12,87	248,56	139,51
3		20,83	11,59				
	10,84			22,02	10,82	238,70	117,29
4		23,21	10,05				
					Tổng	720,16	419,00

Hình N14: *Bảng Tổng hợp khối lượng*

Phụ lục 4

PHẦN MỀM TÍNH KẾT CẤU SAP 2000

Version 7.42 – 138 MB – Dùng với AutoCAD 2000

Đây chỉ là giới thiệu sơ lược cách sử dụng, chủ yếu để bước đầu làm quen, trên cơ sở đó dễ hiểu hơn khi đọc thêm sách hướng dẫn chi tiết.

I. KHÁI QUÁT CHUNG

1. Thuật toán – Quy trình thiết kế

Phương pháp tính toán theo phần tử hữu hạn.

Tính kết cấu SAP được lập theo các quy trình thiết kế kết cấu bê tông của:

- Viện bê tông Mỹ **ACI 318-99** (American Concrete Institute)

- Hội Cầu đường Mỹ **AASHTO LRFD 1997** (American Association of State Highway and Transportation Officials)

- Hội Tiêu chuẩn Canada **CSA-A23.3-94** (Canadian Standards Association)

- Viện Tiêu chuẩn Anh **BS 8110-85 R1989** (British Standards Institution)

- Hội Tiêu chuẩn Châu Âu **EUROCODE 2** (CEN 1992 – European Committee for Standardization)

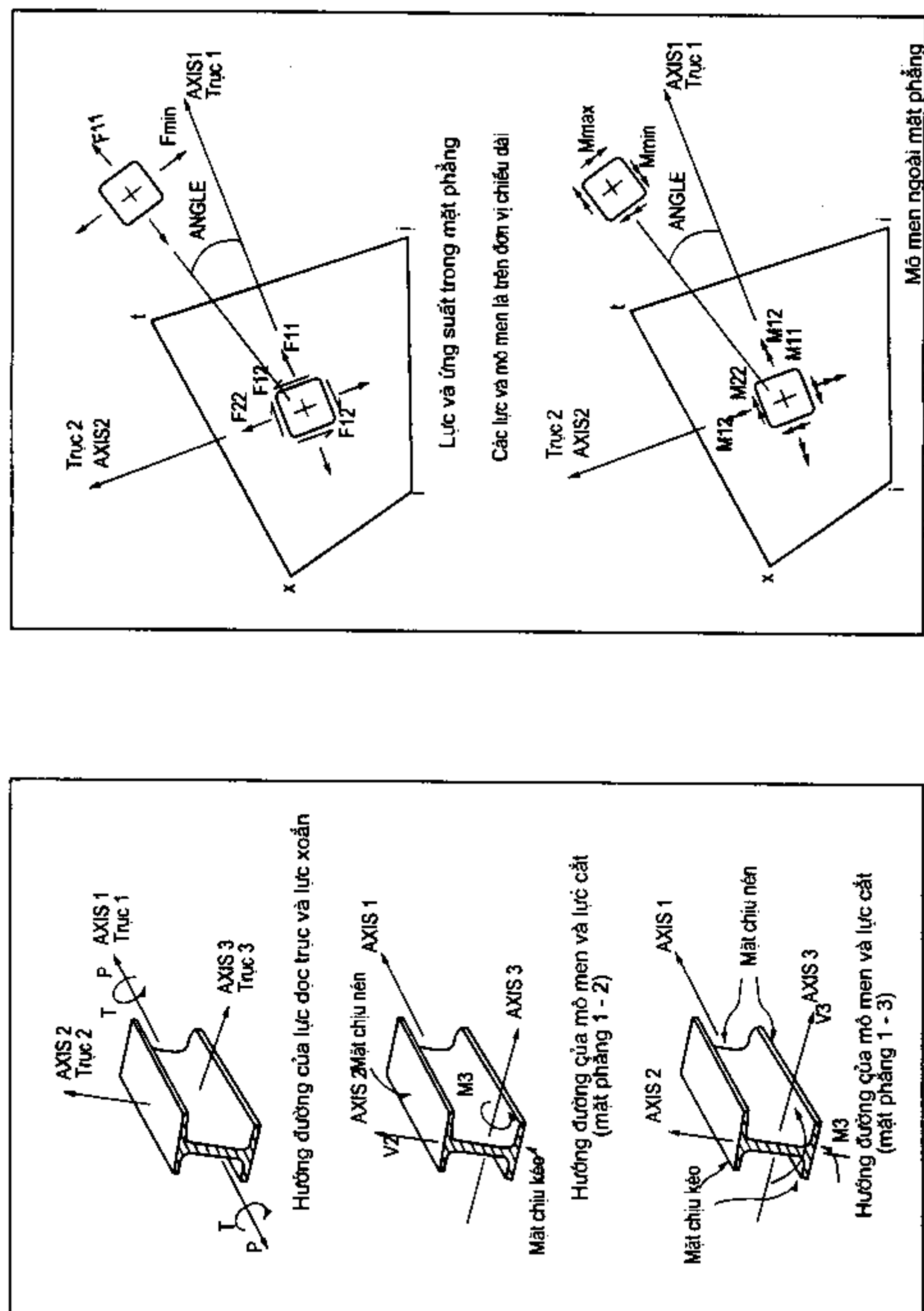
- Tiêu chuẩn New Zealand **NZS 3101-95** (New Zealand Standards)

Hướng dẫn chi tiết tính SAP đọc trong Folder **Manual.** Tuy nhiên, trước hết nên đọc *Quick Tutorials* và xem phần thực hành dưới dạng camera trong Folder **Tutorial**: Vào *Computers and Structures* → *SAP2000 Nonlinear* → *Tutorial* → nhấn *Scplayer* → *Screen Camera* → *Untitled- Lotus S...*→ nhấn biểu tượng *Open File* → ra bảng các danh mục ví dụ → giả sử muốn chạy *Bridge1a.scm*, nhấn vào đó → nhấn ▸ *Play (SPACBAR)* → máy chạy tự động các bước hướng dẫn dưới dạng camera, kèm theo lời hướng dẫn.

2. Hệ trục toạ độ

Phân loại bài toán kết cấu theo hệ toạ độ thường chia 2 loại: bài toán phẳng và kết cấu không gian. Điều này được thể hiện qua hệ toạ độ chung X, Y, Z, hệ toạ độ địa phương trục 1, 2, 3. Khi tính kết cấu chỉ đặt theo 2 chiều gọi là bài toán phẳng, theo 3 chiều là bài toán không gian. Tuy nhiên, ngay cả với bài toán phẳng, vẫn

phải dùng cả 3 chiều để gọi tên các nội lực: chịu nén, chịu cắt, moment, chịu xoắn. Điều này được thể hiện trong SAP như Hình S1.



Hình S1: Các ký hiệu chiều nội lực trong SAP

Bài toán phẳng thường chọn hệ toạ độ X (ngang) Z (đứng) như Hình S2. Tương ứng số hiệu X=1, Z=2. Do vậy moment uốn dầm phẳng trong mặt XZ xoay quanh trục Y gọi là M3.

3. Hệ đơn vị

Hệ đơn vị có thể theo hệ *inch* hoặc *mét* (M). Tóm tắt như sau:

	Chiều dài	*Trọng lượng*	*Lực*	*Ứng suất*
Hệ inch	in, feet (ft)	pound (lb), Kip	pound (lb), Kip	Psi
Hệ M	cm, m	Kg, Ton (Tấn)	KN, Kgf, Ton	Kgf/cm^2, T/m^2

Ở góc dưới màn hình chỉ ghi ngắn gọn như: *lb-in, lb-ft, Kip-in, Kip-ft, N-mm, N-m, KN-mm, KN-m, Ton-mm, Ton-m, Kgf-mm, Kgf-cm, Ton-cm v...v...*

Đơn vị lực *(Kgf)* Việt Nam còn quen viết là *Kgl* hoặc *kG*.

4. Chỉ tiêu cơ lý vật liệu.

Khi thiết kế một kết cấu công trình, việc đầu tiên là xác định chủng loại vật liệu (phổ thông nhất là bê tông, thép), kèm theo đó là các chỉ tiêu cơ lý (mác bê tông, mác thép, cường độ chịu kéo, nén, cắt v...v...). Đây là vấn đề rất quan trọng, vì khi ta nhập số liệu về chỉ tiêu cơ lý vật liệu không chính xác, kết quả tính toán sẽ sai lệch rất nhiều.

Các chỉ tiêu cơ lý lại thường chia 2 loại số liệu:

- Chỉ tiêu kiểm tra, tính toán thực tế
- Chỉ tiêu cho phép

Sau đây là một số quy định chung.

A. Bê tông

a) Mác bê tông: được định nghĩa là cường độ chịu nén dọc trục của mẫu bê tông với *kích thước quy định*, sau 28 ngày bảo dưỡng.

Kích thước quy định mẫu bê tông thí nghiệm tuỳ theo từng nước. Với Việt Nam trước đây, mẫu chịu nén là mẫu hình lập phương 20x20x20 cm, sau lại đổi là 15 × 15 × 15 cm. Theo AASHTO (Mỹ) là mẫu trụ tròn Φ 15 cm, cao 30 cm. Khi dùng mẫu khác quy định, phải chuyển đổi về mẫu quy định qua hệ số đổi cường độ. Tuy nhiên, việc chuyển đổi này khó bảo đảm chính xác, nhất là hiện nay thường dùng phụ gia, tốt nhất là thí nghiệm mác bê tông theo đúng kích thước mẫu quy định.

b) Các chỉ tiêu cơ lý thông dụng của bê tông:

- Cường độ chịu nén (σ_b, Kgf/cm^2): là cường độ chịu nén đúng tâm. Cường độ chịu nén đúng tâm cho phép của bê tông M.200 là $[\sigma_b] = 200\ Kgf/cm^2$.

- Cường độ chịu nén khi uốn (σ_u, Kgf/cm^2): là cường độ phần chịu nén của bê tông khi dầm chịu uốn. Thí dụ cường độ chịu nén khi uốn cho phép của bê tông M.200 là $[\sigma_u]$ = 90 Kgf/cm^2.

- Cường độ chịu kéo khi uốn (σ_{ku}, Kgf/cm^2) : là cường độ chịu kéo khi dầm chịu uốn, thí nghiệm trên mẫu chuẩn 15x15x60 cm. Thí dụ cường độ chịu kéo khi uốn cho phép của bê tông M.200 là $[\sigma_{ku}]$ = 7,50 Kgf/cm^2.

- Mô đun đàn hồi chịu nén của bê tông (E_b, Kgf/cm^2) : nói chung mô đun đàn hồi của vật liệu được xác định theo công thức :

$$E = \sigma_b / \delta \qquad \text{(P4-1)}$$

$$\delta = \Delta h / h \qquad \text{(P4-2)}$$

σ – cường độ chịu nén (hoặc kéo). Với bê tông, khi xác định mô dun đàn hồi chịu nén thường dùng trị số mác bê tông (như M.200 có $\sigma = \sigma_b$ = 200 Kgf/cm^2).

δ - độ co (hoặc dãn dài) tương đối của mẫu chịu nén (hoặc kéo) với mẫu có chiều cao h.

Δh - độ co (hoặc dãn dài) của mẫu. Với bê tông M.200 là ở thời điểm chịu lực nén 200 Kgf/cm^2.

Tóm tắt một số chỉ tiêu cơ lý của bê tông như Bảng 4-1 sau để tham khảo. Các trị số này, tuỳ kết cấu chuyên ngành (như kết cấu nhà, kết cấu mặt đường...) đôi khi có quy định khác nhau.

Bảng P4-1. Một số chỉ tiêu cơ lý của bê tông

Mác bê tông	150	200	250	300	350	400	500
$[\sigma_b]$, Kgf/cm^2	150	200	250	300	350	400	500
$[\sigma_u]$, Kgf/cm^2	65	90	110	130	155	175	-
$[\sigma_{ku}]$, Kgf/cm^2	6,00	7,50	8,30	10,00	11,00	-	-
E_b , 10^5 Kgf/cm^2	2,30	2,65	2,90	3,15	3,30	3,50	3,80

Ghi chú Bảng P4-1: *- Hệ số Poisson của bê tông (Poisson's Ratio) μ = 0.2*

- Trọng lượng đơn vị thể tích bê tông không cốt thép 2.2 T/m^3, bê tông cốt thép 2.5 T/m^3.

B. Cốt thép

a) Phân loại thép

Hiện có nhiều cách phân loại, gọi tên Mác thép,

+ Việt Nam chia 4 nhóm: CI, CII, CIII, CIV

+ Liên Xô cũ chia 4 nhóm: AI, AII, AIII, AIV. Nhóm AI quen gọi là mác CT3, thép trơn. Nhóm AII quen gọi là mác CT5, thép gai.

b) Các chỉ tiêu cơ lý thông dụng của thép

Tóm tắt một số chỉ tiêu cơ lý thông dụng của thép như bảng P4-2 để tham khảo.

Bảng P4-2. Một số chỉ tiêu cơ lý của thép

Nhóm cốt thép	Giới hạn chảy, gf/cm^2	Cường độ giới hạn, Kgf/cm^2	Độ dãn dài tương đối, δ, %	Cường độ chịu kéo cho phép Ra, Kgf/cm^2	Cường độ chịu kéo khi uốn cho phép Ru, Kgf/cm^2	Mô đun đàn hồi, Et. 10^6, Kgf/cm^2
AI (CT3)	2400	3800	25	2100	1600	2,0-2,1
AII (CT5)	3000	5000	19	2700	-	2,0-2,1
AIII	4000	6000	14	-	-	2,0-2,1
AIV	6000	9000	6	-	-	2,0-2,1

Ghi chú Bảng P4-2: *- Trị số mô đun đàn hồi Et = (2,0 – 2,1) là trị số trung bình cho các loại thép thông thường. Với thép cường độ cao dùng cho cầu ứng suất trước, trị số này do hãng sản xuất thép cung cấp.*

- Hệ số Poisson của thép (Poisson's Ratio) $\mu = 0.24 - 0.28$.

- Trọng lượng đơn vị thể tích thép 7,8 T/m^3.

II. VÍ DỤ S-1: TÍNH DẦM LIÊN TỤC

Tính moment và cốt thép dầm bê tông cốt thép liên tục, tiết diện ngang 15x35 cm, 2 nhịp 2xl = 2x400 cm, chịu lực tập trung P = 3.740 Kgf đặt ở giữa nhịp 1, chịu tải trọng phân bố hình thang ở nhịp 2, (300-500Kg/m trên chiều dài 1m, cách cuối dầm 1m). Bê tông M.200, cốt thép gai AII.

1. Vào *New File* ra *Coordinate System Definition* → OK → *XY Plane*, nhấn xz trên thanh công cụ ra trục toạ độ ngang X, đứng Z. Xem Hình S2

Chọn đơn vị ở góc cuối màn hình, thí dụ Kgf-cm

Muốn xoá bỏ lưới trên hình S2 cho dễ nhìn, nhấn F7.

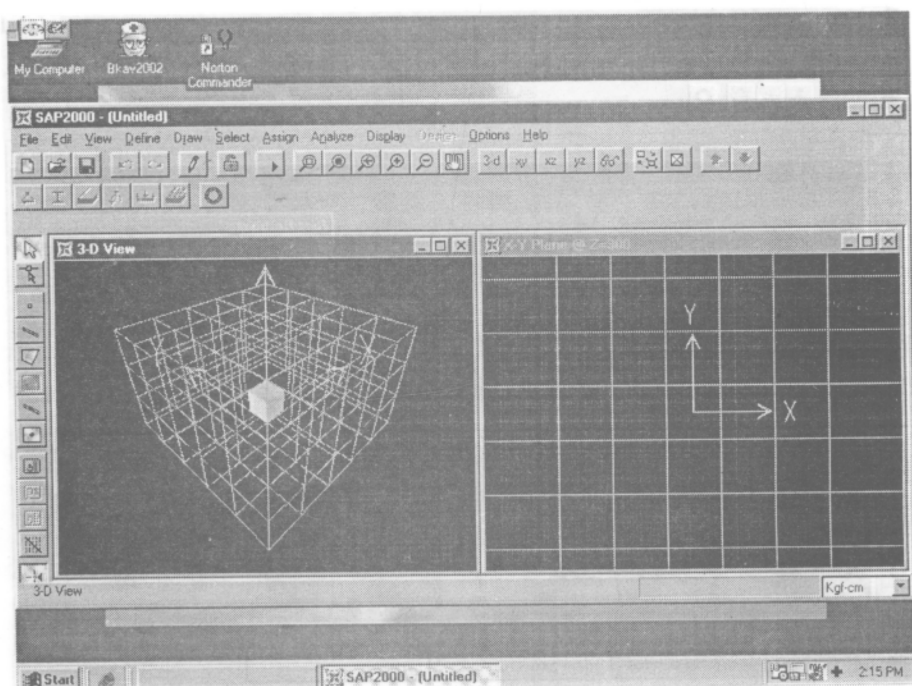
2. Nhấn phần tử (o) trên thanh công cụ → nhấn trỏ lên màn hình, hiện ra điểm đầu dầm (o) trên màn hình. Nhấn *Esc* kết thúc lệnh.

Nhấn trỏ vào (o) để đánh dấu. Nhấn *Ctrl+C*, *Ctrl+V* ra hộp thoại *Paste Coordinate*. Tại X nhập trị số 400 → OK → ra đầu dầm thứ hai. Xem Hình S3.

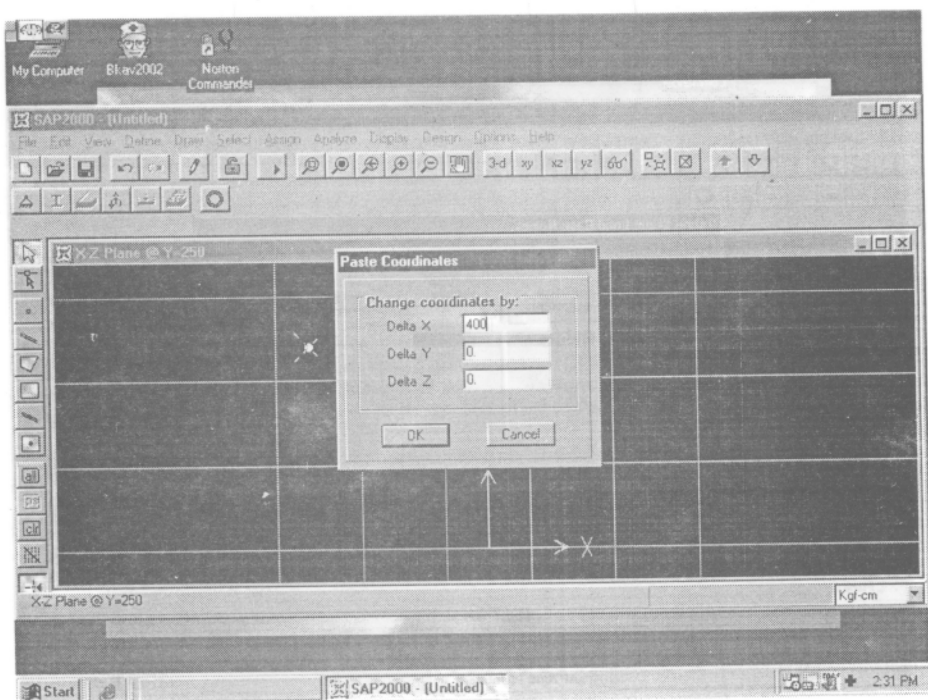
Lặp lại như trên ra đầu dầm thứ ba (o)

3. Nhấn (\) trên thanh công cụ → vẽ nối 3 điểm đầu dầm → Esc kết thúc lệnh

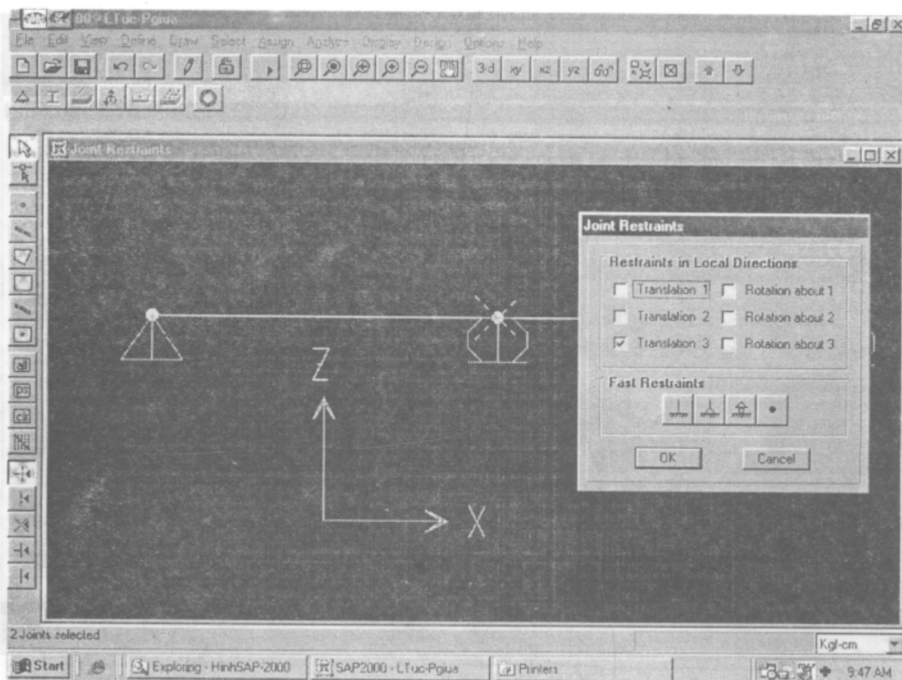
4. Đánh dấu đầu dầm, nhấn (Δ) ra *Joint Restraints*. Nhấn vào gối cố định (hoặc di động) tuỳ ý.



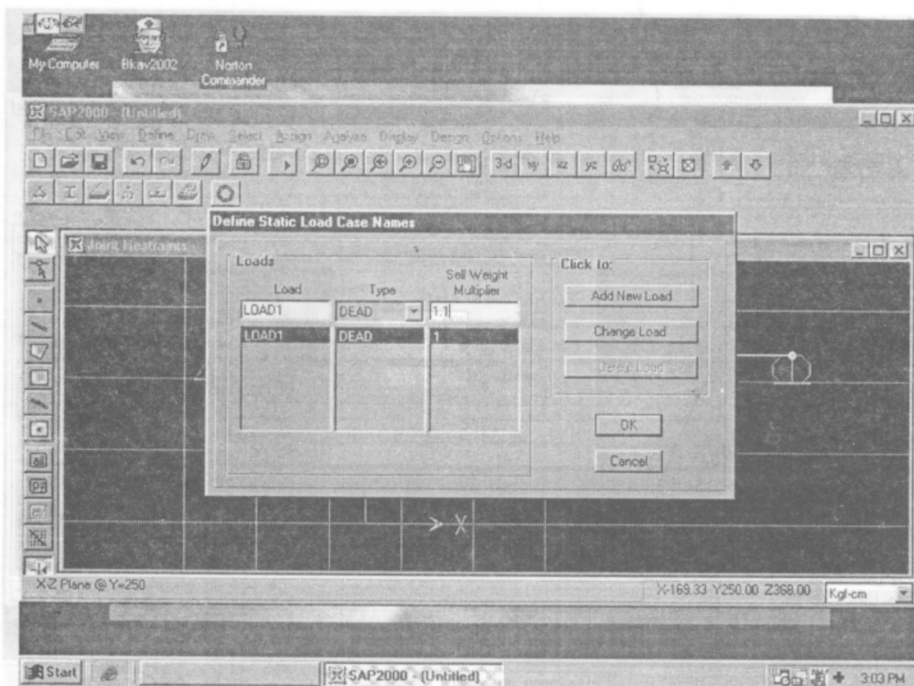
Hình S2: *Hệ trục toạ độ*



Hình S3: *Hộp thoại* Paste Coordinate - *Xác định đầu dầm thứ hai*



***Hình S4:** Hộp thoại* Joint Restraints *- Xác định gối dầm*



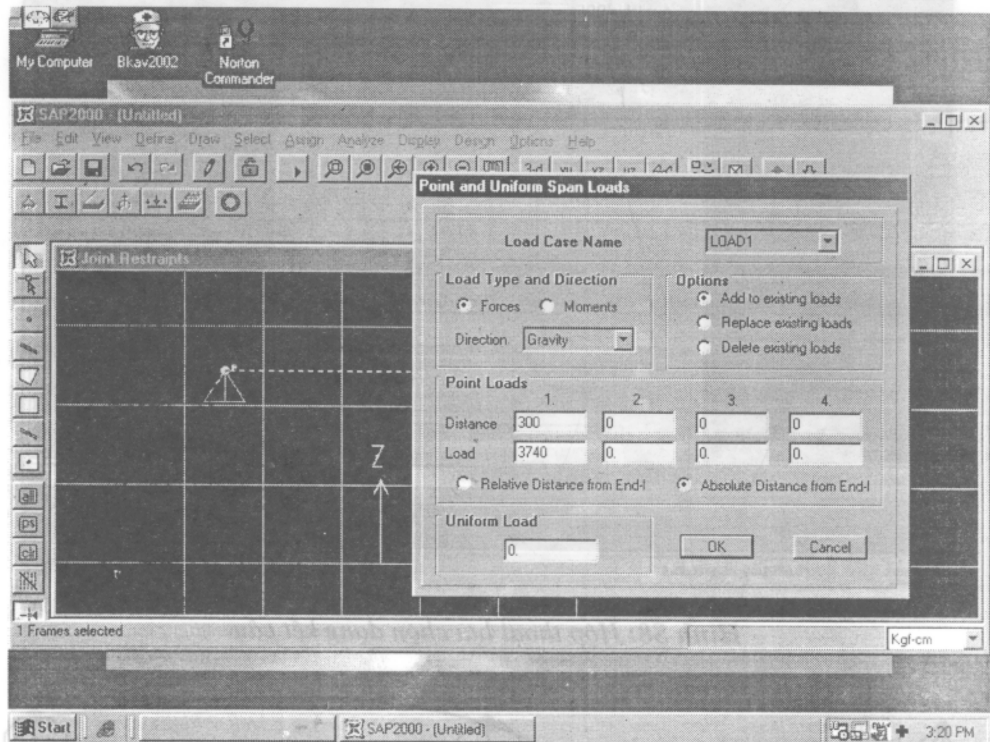
***Hình S5:** Đặt tĩnh tải (trọng lượng bản thân dầm)*

5. Vào menu *Define* → *Static load case* ra hình S5 để đặt tĩnh tải, loại trọng lượng bản thân dầm ở cột *Self Weight Multiplier*. Nếu có xét đến trọng lượng bản thân, nhập số 1.1 (là hệ số vượt tải). Nếu không xét đến trọng lượng bản thân, nhập số 0 → OK

6. Đánh dấu dầm nhịp 1 (ra nét đứt ------), nhập tải trọng tập trung P. Nhấn ↓↓↓ trên thanh công cụ để đặt các loại tải trọng khác trên dầm, ra Hình S6. Đánh dấu mục *Absolute Distance from End-1*. Nhập số liệu khoảng cách *(Distance)* 200 cm và tải trọng *(Load)* 3740 Kgf → ra Hình S7.

Đánh dấu dầm nhịp 2, nhập tải trọng phân bố hình thang. Vào *Assign* → *Frame Static Load* → *Trapezoidal* → đánh dấu mục *Absolute Distance from End-1* → mục *Direction* chọn *Gravity* → nhập số liệu tương tự hình S19 → OK ra Hình S7.

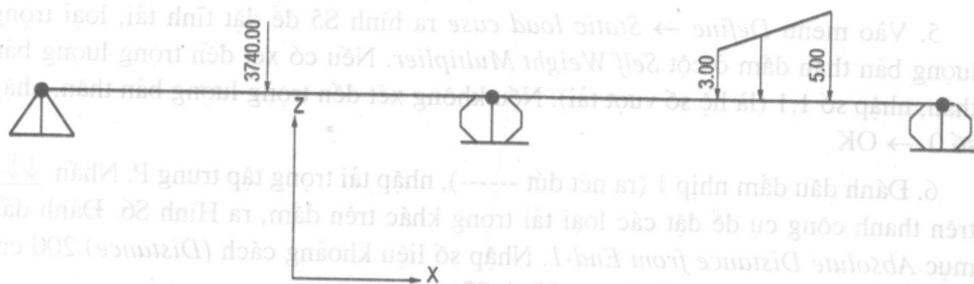
Muốn *in hình*, nhấn Ctrl+G.



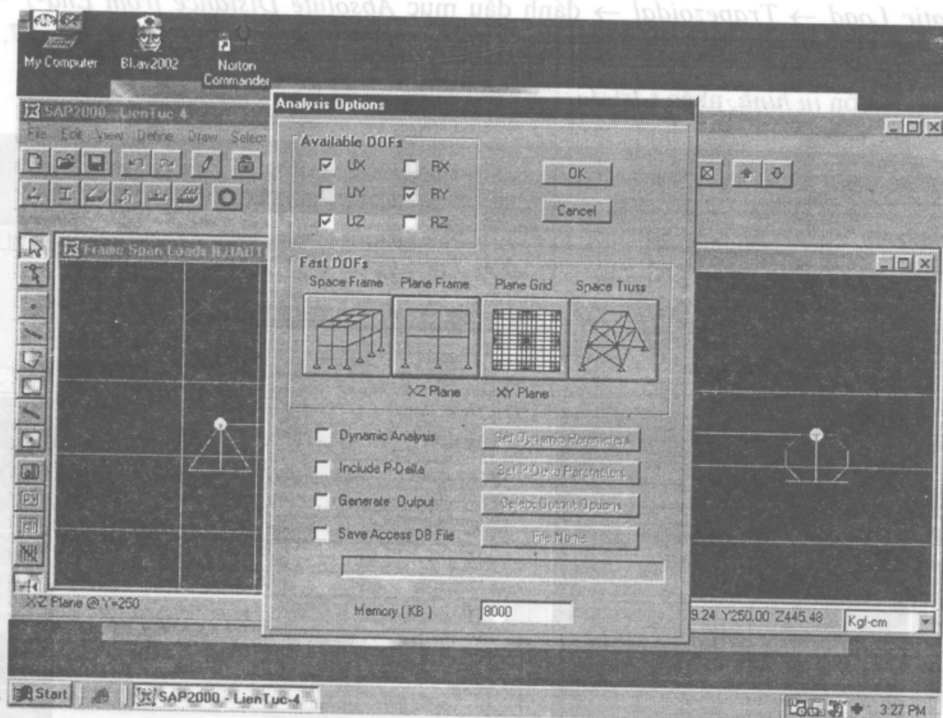
Hình S6: *Hộp thoại đặt các loại tải trọng tập trung, phân bố khác*

7. Vào *Analyze* → *Set Option* ra Hình S8 → Nhấn *XZ Plane* → OK

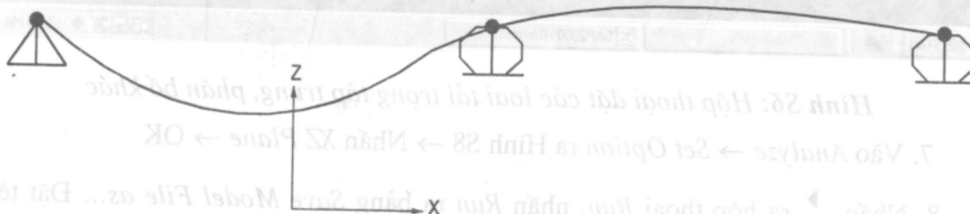
8. Nhấn ▸ ra hộp thoại *Run*, nhấn *Run* ra bảng *Save Model File as*... Đặt tên File theo đường dẫn (thí dụ C:\ Tinh-SAP \ DamAC). Nhấn *Save* → máy chạy tính toán → *Analysis Complete* → OK, ra Hình S9.



Hình S7: *Sơ đồ đặt tải*

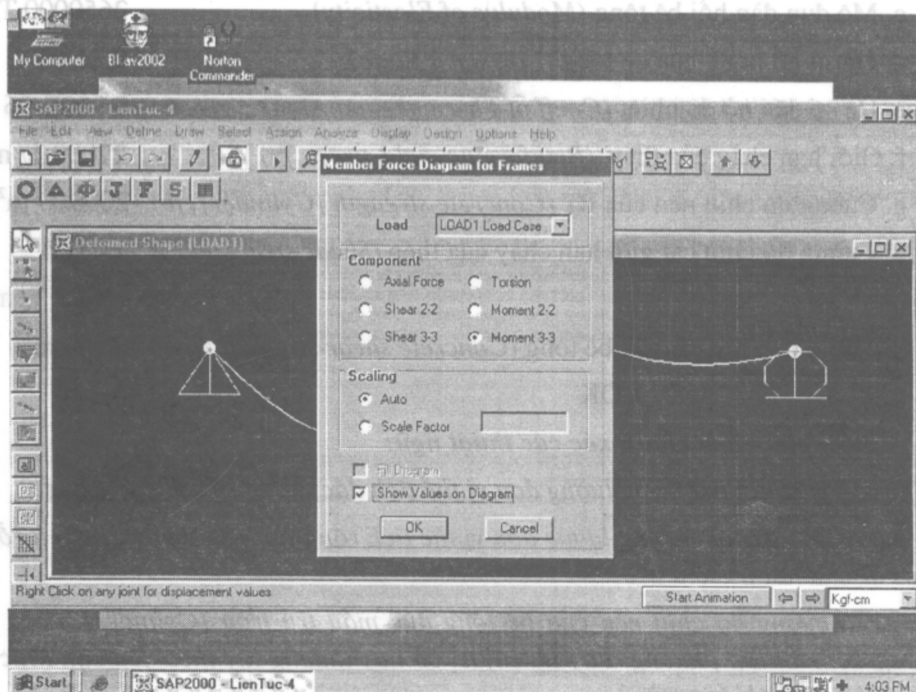


Hình S8: *Hộp thoại lựa chọn dạng kết cấu*

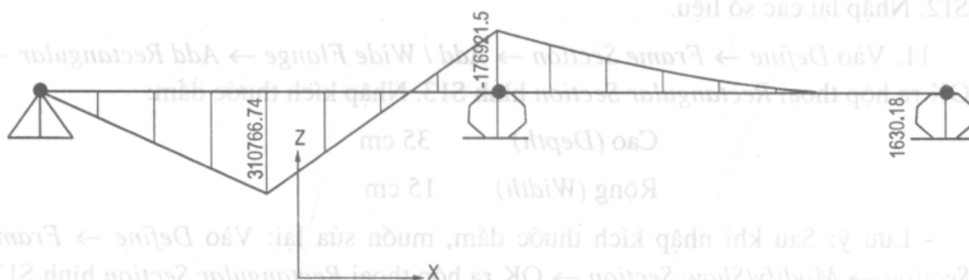


Hình S9: *Biến dạng dầm liên tục*

9. Vào *Display* → *Show Element Forces/ Stresses* → *Frame* ra hộp thoại *Member Forces Diagram For Frames* như Hình S10. Chọn *Moment 3-3, Show Value* → OK ra Hình S11. (Lý do chọn moment 3-3 xem hình S1).



Hình S10: *Hộp thoại Member Forces Diagram For Frames*



Hình S11: *Biểu đồ moment*

10. Nhấn biểu tượng khoá trên thanh công cụ để mở khoá, hình S11 sẽ mất biểu đồ moment, chỉ còn sơ đồ dầm.

Vào *Define* → *Material* → *Add New Material / Show Material ra* hộp thoại *Material Property Data* như Hình S12. Nhập các số liệu như sau, chú ý chuyển đổi thống nhất T-m hoặc Kgf-cm :

a.Khối lượng đơn vị thể tích bê tông *(Mass per unit Volume)* 2.5 T/m^3

b.Trọng lượng đơn vị thể tích bê tông *(Weight per unit Volume)* 2.5 T/m^3

c. Mô đun đàn hồi bê tông *(Modulus of Elasticity)* 2650000 T/m^2

d. Hệ số nở hông của bê tông *(Poisson's Ratio)* 0.2

e. Hệ số dãn nở do nhiệt *(Coeff of Thermal expansion)* 9.900E-06

f. Giới hạn chảy của thép *(Reinforcing yield stress, fy)* 27000 T/ m^2

g. Cường độ chịu nén của BT *(Concrete strength [Cylinder], fc)* 2000 T/ m^2

h. Cường độ chịu cắt giới hạn chảy của thép *(Shear steel yield stress, fys)*

16000 T/ m^2

i. Cường độ chịu cắt của bê tông *(Concrete shear strength, fcs)* 75 T/ m^2

OK → OK

- Lưu ý phân biệt chính xác các thuật ngữ:

(a) có ý nghĩa như trọng lượng đơn vị thể tích đặc, về trị số bằng tỷ trọng.

(b) có ý nghĩa như trọng lượng đơn vị thể tích rỗng. Vì bê tông đặc (hay ít rỗng) nên coi như hai trị số a = b

(g) là cường độ chịu nén của bê tông đúc mẫu trụ tròn [Cylinder], khác với cường độ mẫu lập phương. Tuy nhiên, trị số này thường lấy theo yêu cầu Mác Bê tông thiết kế dự kiến trước.

- Sau khi nhập chỉ tiêu cơ lý vật liệu như trên, muốn sửalại: Vào *Define → Material → Modify/ Show Material ra* hộp thoại *Material Property Data* như Hình S12. Nhập lại các số liệu.

11. Vào *Define → Frame Section → Add / Wide Flange → Add Rectangular →* OK ra hộp thoại *Rectangular Section* hình S13. Nhập kích thước dầm:

Cao *(Depth)* 35 cm

Rộng *(Width)* 15 cm

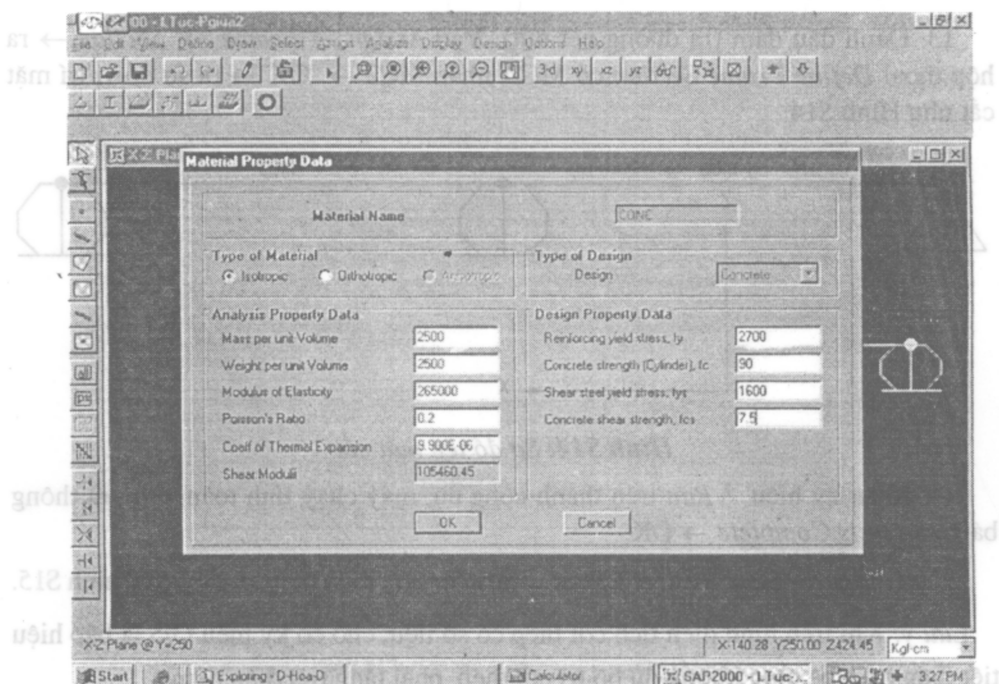
- Lưu ý: Sau khi nhập kích thước dầm, muốn sửa lại: Vào *Define → Frame Section → Modify/Show Section →* OK ra hộp thoai *Rectangular Section* hình S13. Nhập lại kích thước dầm

12. Nhấn tiếp vào *Reinforcement* ra hộp thoại *Reinforcement Data*, chọn *Beam*, nhập bề dày lớp bảo vệ cốt thép

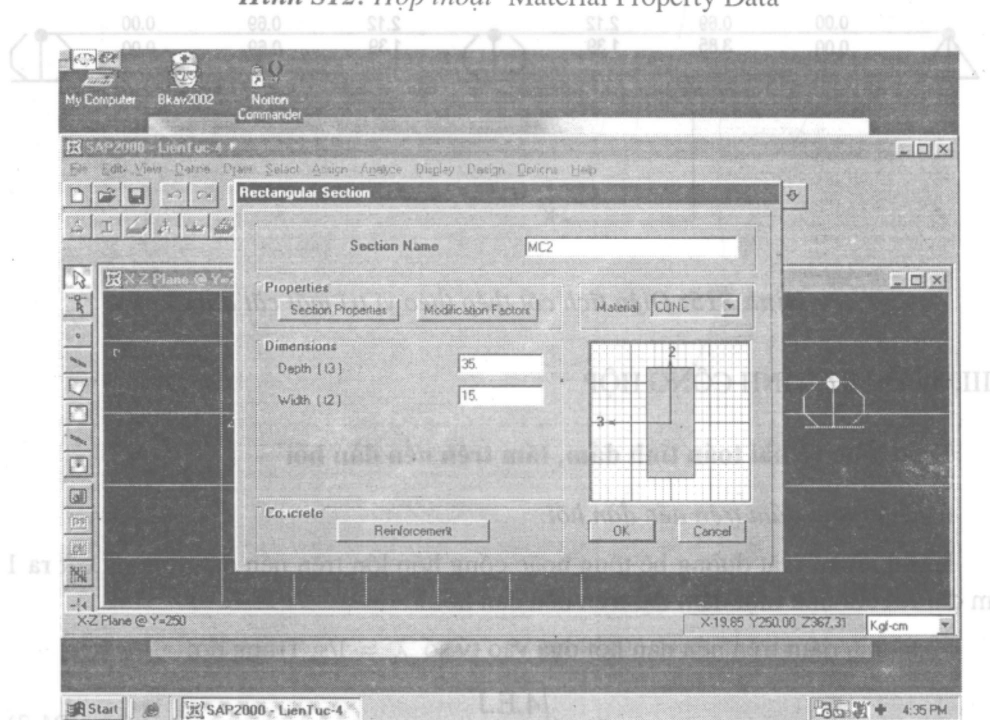
Đỉnh *(Top)* 3 cm

Đáy *(Bottom)* 3 cm

Mục *Section name* đặt tên mặt cắt MC2 → OK

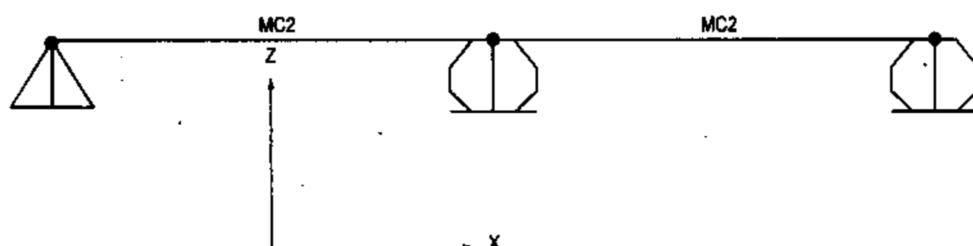


Hình S12: Hộp thoại Material Property Data



Hình S13: Hộp thoại Rectangular Section

13. Đánh dấu dầm (ra đường nét đứt). Vào *Assign* → *Frame* → *Section* → ra hộp thoại *Define Frame Section* có MC2, nhấn MC2 → OK → ra sơ đồ vị trí mặt cắt như Hình S14

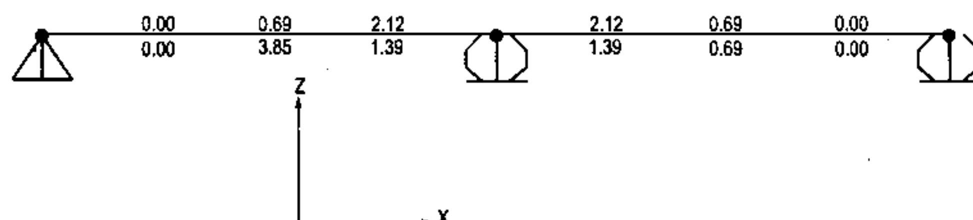


Hình S14: *Sơ đồ tên mặt cắt*

14. Nhấn ký hiệu ▸ *Run* trên thanh công cụ, máy chạy tính toán xong ra thông báo: *Analysis Complete* → OK.

Vào *Design* → *Start Design/ Check of Structure* ra diện tích cốt thép như hình S15.

Lưu ý: Khi trên hình diện tích cốt thép có số liệu, chỗ có ký hiệu O/S là báo hiệu tiết diện dầm bê tông không đủ bố trí cốt thép, phải tăng tiết diện dầm.



Hình S15: *Diện tích cốt thép theo vị trí mặt cắt dầm*

III. VÍ DỤ S2: TÍNH CỐNG HỘP

1. Sơ lược về bài toán tính dầm, tấm trên nền đàn hồi

a. Phân loại dầm trên nền đàn hồi

Khi thiết kế mặt đường bê tông hoặc cống hộp lớn trên nền đất, thường cắt ra 1 m dài rồi coi như một dầm đặt trên nền đàn hồi.

Phân loại dầm trên nền đàn hồi dựa vào tỷ số $\lambda = l/s$. Trong đó:

$$s = \sqrt[4]{\frac{4.E.J}{b.K_s}} \qquad \text{(P4-3)}$$

l (cm) - chiều dài dầm, b =100 cm – bề rộng dầm.

K_s (Kgf/cm^3) – hệ số nền.

E (Kgf/cm^2) – mô đun đàn hồi vật liệu dầm

J (cm^4) – mô men quán tính tiết diện dầm. Với dầm chữ nhật $J = bh^3/12$

Khi $\lambda = l/s < 0{,}75$ gọi là dầm cứng (có thể bỏ qua lực uốn của dầm)

$0{,}75 < \lambda < 3$ gọi là dầm ngắn

$\lambda > 3$ gọi là dầm dài.

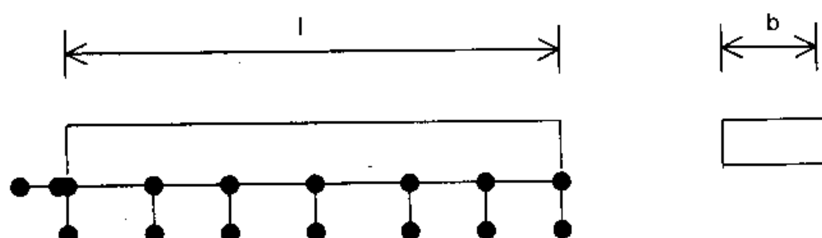
b. Hệ số nền K_s

Dầm trên nền đàn hồi hiện nay có nhiều giả thuyết, sơ đồ tính toán khác nhau, nhưng đơn giản nhất là theo mô hình Winkler: độ lún y tại một điểm đã cho của nền không phụ thuộc vào độ lún của những điểm khác và tỷ lệ với áp lực p tại điểm ấy, tức là:

$$p = K_s.y \qquad \text{(P4-4)}$$

Trong đó K_s là hệ số nền (Kgf/cm^2/cm = Kgf/cm^3) lấy bằng hằng số. Tham khảo trị số sau, theo P. L. Pasterσnac :

- Với đất rất yếu : $K_s = 0{,}3 - 1{,}0$ Kgf/cm^3
- Với đất yếu : $K_s = 1 - 3$
- Với đất chặt vừa : $K_s = 3 - 8$
- Với đất rất chặt : $K_s = 8 - 20$
- Với đất đá cứng : $K_s = 400 - 800$



***Hình S16:** Sơ đồ nguyên lý tính dầm trên nền đàn hồi*

c. Gán độ cứng lò xo

Độ cứng lò xo của dầm trên nền đàn hồi, cọc, tường cừ được tính theo công thức:

$$K_{lx} = K_s.b.l_s, \qquad \text{Kgf/cm}^2 \qquad \text{(P4-5)}$$

Trong đó: K_{lx} là thành phần độ cứng lò xo theo phương vuông góc với dầm hoặc cọc cừ.

K_s - hệ số nền, Kgf/cm^3;

b - bề rộng nền, cm;

l_s - khoảng cách giữa các lò xo, cm.

2. Ví dụ S2

Ví dụ này chủ yếu giới thiệu sơ lược cách dùng SAP tính toán bài toán dầm trên nền đàn hồi, không đề cập chi tiết mọi tình huống tổ hợp tải trọng.

a. Yêu cầu tính toán

Tính moment và cốt thép cống hộp bê tông cốt thép tiết diện thông thuỷ $B \times H = 3{,}00 \times 2{,}80$m. Kích thướctiết diện ngang dự kiến như hình S17. Bê tông M.300, cốt thép gai AII.

Cống chôn trong đất á cát có dung trọng tự nhiên $\gamma_w = 1{,}90\ T/m^3$, góc nội ma sát $\varphi = 20^o$, lực dính $C = 0{,}50\ T/m^2$. Góc ma sát giữa đất và thành cống $\delta = 19^o$.

Nền đất đáy cống thuộc loại đất yếu, có hệ số nền $K_s = 0{,}60\ Kgf/cm^3$.

Tải trọng trên cống gồm 30cm mặt đường phía trên, xe HK80 được quy về lớp đất tương đương cao h_o (với cùng loại đất á sét nêu trên), tải trọng nước chảy đầy cống.

$$h_o = \frac{\sum P}{B.L.\gamma_w} \qquad \text{(P4-6)}$$

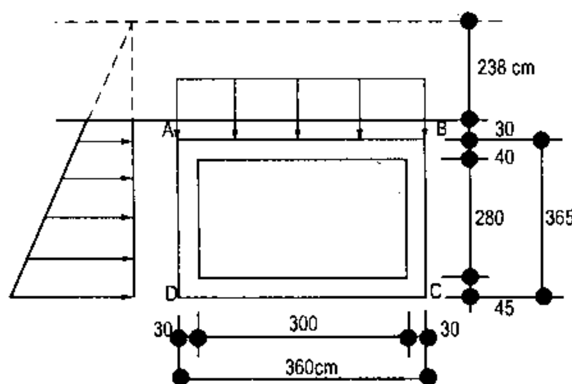
Trong đó :

ΣP = 80T, là tải trọng xe HK80

B = 2,7 + 0,8 = 3,5 m, là bề rộng phủ bì lốp xe

$L = 1{,}2 \times 3 + 0{,}2 + 2.\Delta H$ = 5,04 m, là chiều dài phân bố tải trọng, trên cơ sở chiều dài phủ bì lốp xe và chiều dày mặt đường $\Delta H = 0{,}62$ m.

$$h_o = \frac{80}{3{,}5 \times 5{,}04 \times 1{,}9} = \frac{4{,}53}{1{,}9} = 2{,}38\text{m}$$



Hình S17: *Sơ đồ tính toán cống hộp 3,00 × 2,80m*

- Toàn bộ tải trọng phân bố đặt trên đỉnh cống, theo 1m cắt dọc cống, gồm: tải trọng xe HK80 (4.53 T/m) + tĩnh tải mặt đường (0,66T/ m) + trọng lượng bản thân cống (3.29 T/ m) + nước đầy cống (2.33 T/ m) = 10.81 T/ m = 108.1 Kgf/cm

- Tải trọng ngang tác dụng lên thành cống (hình thang) gồm áp lực đất, (có tính cả chiều cao cột đất tương đương 2,38m do xe HK80 gây ra), và áp lực nước đầy ra khi cống chảy đầy nước.

b. Các bước tính toán

1. *Vẽ sơ đồ tiết diện cống theo đường tim bề dày cống.* Như vậy bề rộng cống l = 330cm, chiều cao h = 322.5 cm. Thao tác tương tự bước 1, 2, 3 ví dụ S1.

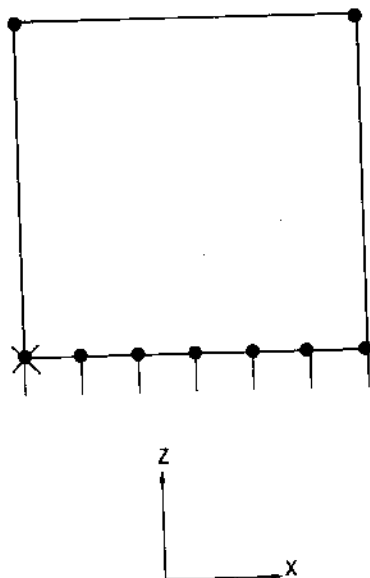
2. Đặt một gối chống chuyển vị ngang theo phương X (trục số 1) : đánh dấu điểm D, nhấn (Δ) ra Joint Restraints (như Hình S4) → đánh dấu vào Translation 1 → OK.

3. *Chia đoạn đặt gối đàn hồi:* giả sử chia đáy cống (đoạn CD) thành 6 đoạn. Đánh dấu mặt đáy, vào *Edit → Divide Frames → Divide Selected Frame → Divide into 6*. Dùng *Last/First Ratio 1* có nghĩa là chia đoạn bằng nhau → OK → ra hình như Hình S18.

4. *Gán lò xo:* Đáy cống CD chia 6 đoạn, chiều dài mỗi đoạn l_x = 55 cm. Áp dụng công thức (P4-5) tính được độ cứng lò xo:

$$K_{lx} = 0,60 \times 100 \times 55 = 3300 \text{ Kgf/cm}$$

Đánh dấu 7 gối chia 6 đoạn mặt đáy cống CD, vào *Assign → Joint → Spring → Joint Springs →* ở mục *Translation 3* khai độ cứng lò xo *3300*, đánh dấu mục *Add to Existing Springs* → OK ra Hình S18.



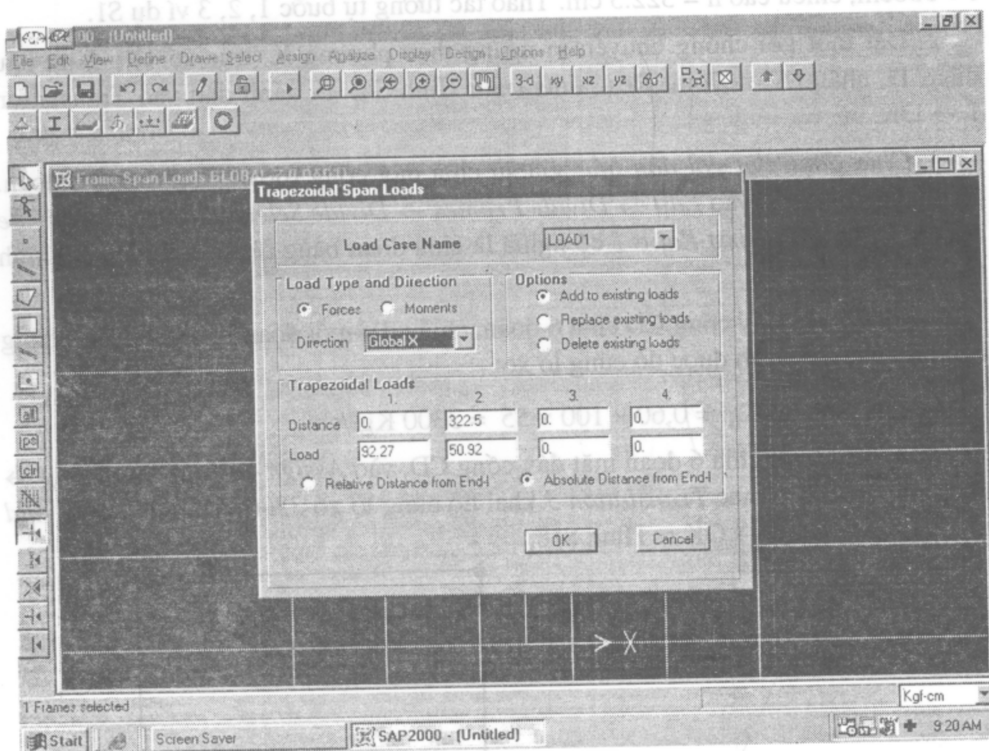
***Hình S18:** Sơ đồ định vị gối tại D và chia lò xo đáy cống*

(Khi muốn kiểm tra lại số liệu độ cứng lò xo đã nhập, vào *Display* → *Show Input Tables* → *Geometry Data* → ra hộp thoại *Display Geometry Data* → đánh dấu vào *Joint Spring Data* → OK → ra bảng, cột K-U3 có trị số 3300).

5. *Chất tải ngang áp lực đất:* có dạng hình thang (Trapezoidal).

- Đầu tiên đánh dấu cạnh cống đoạn AD, trong đó tung độ điểm D = 0, điểm A = 322.5 cm.

Vào Assign → Frame Static Load → Trapezoidal → đánh dấu mục Absolute Distance from End-1 → mục Direction chọn Global X → nhập số liệu như hình S19.

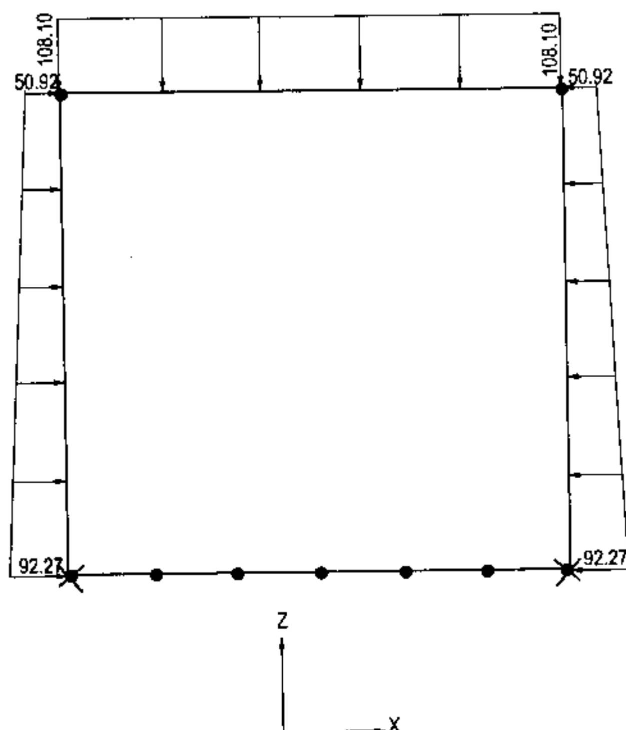


Hình S19: *Nhập số liệu lực ngang hình thang*

- Đánh dấu cạnh cống đoạn BC, trong đó tung độ điểm B = 0, điểm C = 322.5 cm. Vào *Assign* và làm tiếp như trên. Lưu ý trị số lực ngược chiều cạnh AD nên mang dấu âm (-).

5. *Chất tải đứng:* Toàn bộ tải trọng đứng được chất lên mặt trên (cạnh AB). Đánh dấu cạnh AB. Nhấn ↓↓↓ trên thanh công cụ để đặt hoạt tải, ra Hình S6. Vì tải trọng đứng là phân bố đều nên nhập số liệu 108.1 Kgf/cm vào mục *Uniform Load*, mục *Direction* chọn *Gravity* (chiều trọng lực) → OK → ra hình S20.

6. Vào *Analyze* → *Set Option* → nhấn biểu tượng *XZ Plane* (như Hình S8) → OK



***Hình S20:** Sơ đồ đặt tải*

7. Nhấn ▸ *Run* → ra bảng *Save Model File as*... Đặt tên File theo đường dẫn (thí dụ C:\TinhSAP\CgHop-3×2.8). Nhấn *Save* → máy chạy tính toán → *Analysis Complete* → OK → ra hình khung cống bị biến dạng.

8. Vào *Display* → *Show Element Forces/ Stresses* → *Frame* ra hộp thoại *Member Forces Diagram For Frames* như hình S10. Chọn *Moment 3-3, Show Value* → OK ra Hình S21. (Lý do chọn moment 3-3 xem Hình S1).

9. Nhấn biểu tượng khoá trên thanh công cụ để mở khoá, hình S21 sẽ mất biểu đồ moment, chỉ còn sơ đồ dầm.

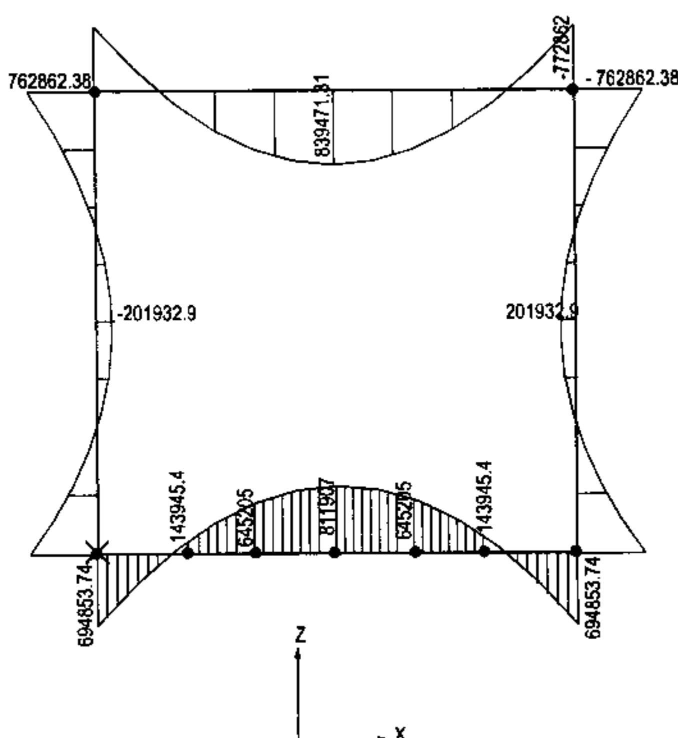
Vào *Define* → *Material* → *Add New Material/Show Material ra* hộp thoại *Material Property Data* như Hình S12. Nhập các số liệu như bước 10 trong Ví dụ S1, nhưng ở đây là bê tông M.300.

a.Khối lượng đơn vị thể tích bê tông *(Mass per unit Volume)* 0,0025 Kg/cm^3

b.Trọng lượng đơn vị thể tích bê tông *(Weight per unit Volume)* 0,0025 Kg/cm^3

c. Mô đun đàn hồi bê tông *(Modulus of Elasticity)* 315000 Kgf/cm^2

d. Hệ số nở hông của bê tông *(Poisson's Ratio)* 0,2



Hình S21: *Sơ đồ moment cống hộp*

e. Hệ số dãn nở do nhiệt *(Coeff of Thermal expansion)* 9,900E-06

f. Giới hạn chảy của thép *(Reinforcing yield stress, fy)* 2700 Kgf/cm^2

g. Cường độ chịu nén của BT *(Concrete strength [Cylinder], fc)* 300 Kgf/cm^2

h. Cường độ chịu cắt giới hạn chảy của thép *(Shear steel yield stress, fys)*

1600 Kgf/ cm^2

i. Cường độ chịu cắt của bê tông *(Concrete shear strength, fcs)* 10 Kgf/cm^2

OK → OK

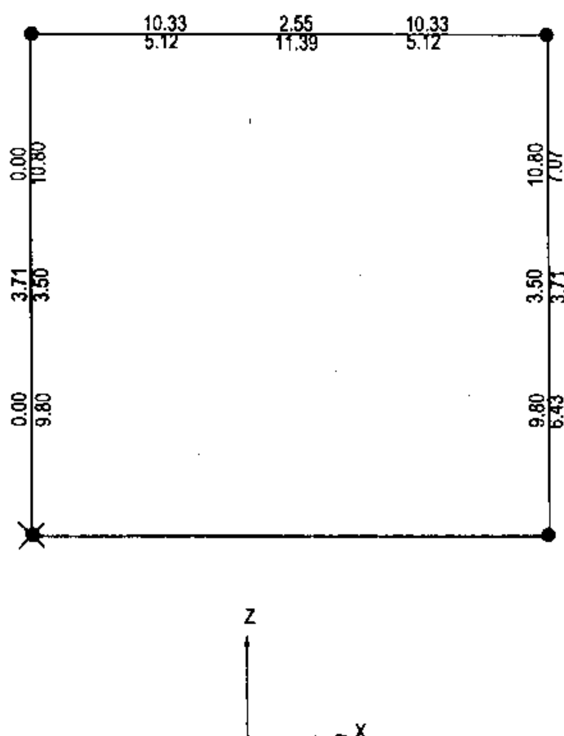
10. *Khai 3 mặt cắt:* MC1 cho đoạn AB, MC2 cho AD, BC ; MC3 cho CD. Vào *Define → Frame Section → Add/ Wide Flange → Add Rectangular →* OK → ra hộp thoại *Rectangular Section* như Hình S13. Làm như bước 11 thí dụ S1.

11. Khai mặt cắt xong, *lần lượt đánh dấu dầm* AB, AD, BC, CD. Vào *Assign → Frame → Section →* ra hộp thoại *Define Frame Section* đã có đủ 3 mặt cắt MC1, MC2, MC3. Nếu đánh dấu đoạn AB, nhấn MC1 → OK ; tiếp theo đánh dấu AD, BC, nhấn MC2 → OK ; cuối cùng đánh dấu CD, nhấn MC3 → OK → ra sơ đồ vị trí các mặt cắt (tương tự hình S14).

12. Nhấn ▸ *Run* → máy chạy tính toán xong ra thông báo *Analysis Complete* → OK

Vào *Design* → *Start Design / Check of Structure* → ra kết quả diện tích cốt thép như Hình S22.

Thực tế số liệu cốt thép hay có trục trặc, ta có thể dựa vào biểu đồ mô men để tính kiểm tra lại.



Hình S22: *Diện tích cốt thép cống hộp*

Phụ lục 5

TRÍCH DỊCH "SỔ TAY THIẾT KẾ BÊ TÔNG"
SAP 2000 – VERSION 7.0
Computers and Structures – CONCRETE DESIGN MANUAL

SỔ TAY THIẾT KẾ BÊ TÔNG

Chương III

THIẾT KẾ THEO TIÊU CHUẨN ACI 318-99

Chương này mô tả chi tiết trình tự thiết kế các kết cấu bê tông khác nhau khi dùng SAP2000 thiết kế theo Tiêu chuẩn của Viện Bê tông Mỹ ACI 318-99 (ACI1999). Các ký hiệu dùng trong chương này xem Bảng III-1

Thiết kế dựa trên các tổ hợp tải trọng do người dùng quy định. Nhưng chương trình cũng cung cấp các tổ hợp tải trọng mặc định thoả mãn yêu cầu thiết kế hầu hết các dạng kết cấu công trình.

SAP2000 cung cấp các lựa chọn thiết kế *Thông thường (Ordinarry)*, *Trung gian (Intermediate)* (như vùng có động đất), và *Đặc biệt (Special)* (như vùng có động đất cao) các kết cấu mô men chống uốn được thiết kế theo yêu cầu động đất cho trước. Chi tiết các thông số dùng cho các hệ kết cấu khác nhau lần lượt được mô tả ở các phần sau.

Hệ đơn vị của Anh SI và MKS được dùng để nhập số liệu. Nhưng chương trình dùng hệ **Inch-Pound-Second**. Để đơn giản, các phương trình trình bày trong chương này đều theo đơn vị **Inch-Pound-Second**, trừ khi có ghi chú khác.

Bảng III-1. Bảng ký hiệu dùng cho Tiêu chuẩn ACI

Ký hiệu	Diễn giải
A_{cv}	Diện tích bê tông dùng để xác định ứng suất cắt, in^2 (sq-in)
A_g	Diện tích toàn bộ bê tông, in^2
A_s	Diện tích cốt thép chịu kéo, in^2
A_s'	Diện tích cốt thép chịu nén, in^2

Bảng III-1 (tiếp theo)

Ký hiệu	Diễn giải
$A_{s(required)}$	Diện tích cốt thép chịu kéo yêu cầu, in^2
A_{st}	Tổng diện tích cốt thép dọc cột, in^2
A_v	Diện tích cốt thép chịu cắt, in^2
a	Chiều cao vùng chịu nén, in
a_b	Chiều cao vùng chịu nén trong điều kiện cân bằng, in
b	Bề rộng cấu kiện, in
b_f	Bề rộng chịu tác dụng của cánh dầm (dầm chữ T), in
b_w	Bề rộng thân dầm (dầm chữ T), in
C_m	Hệ số, phụ thuộc vào độ cong của cột, dùng để tính toán hệ số tăng mô men
c	Chiều cao trục trung hoà, in
c_b	Chiều cao trục trung hoà khi điều kiện cân bằng, in
d	Khoảng cách từ mặt cốt thép chịu nén tới chịu kéo, in
d'	Lớp bảo vệ bê tông tính tới tim cốt thép, in
d_s	Bề dày bản bê tông (dầm chữ T), in
E_c	Mô đun đàn hồi của bê tông, psi (p/in^2)
E_s	Mô đun đàn hồi của cốt thép, psi
f_c'	Cường độ chịu nén tiêu chuẩn của bê tông, psi
f_y	Cường độ giới hạn chảy dẻo của cốt thép khi chịu uốn, psi
f_{ys}	Cường độ giới hạn chảy dẻo của cốt thép khi chịu cắt, psi
h	Kích thước cột, in
I_g	Mô men quán tính của toàn tiết diện mặt cắt bê tông quanh trục trung tâm, không tính đến cốt thép, in^4
I_{se}	Mô men quán tính của cốt thép quanh trục trung tâm của cắt ngang cấu kiện, in^4
k	Hệ số ảnh hưởng của chiều dài
L	Chiều dài tĩnh không, in
M_1	Trị số giảm nhỏ mô men cuối cột, lb-in
M_2	Trị số tăng mô men trong cột, lb-in
M_c	Trị số mô men dùng để thiết kế, lb-in
M_{ns}	Trị số mô men cuối của cấu kiện không lắc lư, lb-in

Bảng III-1 (tiếp theo)

Ký hiệu	Diễn giải
M_s	Trị số mô men cuối của cấu kiện có lắc lư, lb-in
M_u	Trị số mô men tại mặt cắt, lb-in
M_{ux}	Trị số mô men tại mặt cắt quanh trục X, lb-in
M_{uy}	Trị số mô men tại mặt cắt quanh trục Y, lb-in
P_b	Khả năng chịu tải dọc trục trong điều kiện cân bằng chịu kéo, lb
P_c	Cường độ chịu nhiệt cực hạn của cột, lb
P_{max}	Cường độ chịu tải dọc trục lớn nhất cho phép, lb
P_0	Khả năng chịu tải dọc trục tại điểm lệch tâm zero, lb
P_u	Trị số chịu tải dọc trục tại mặt cắt, lb
r	Bán kính cong của mặt cắt cột, in
V_c	Sức chống cắt của bê tông, lb
V_E	Sức chống cắt gây bởi lực động đất, lb
V_{D+L}	Sức chống cắt do tải trọng trên nhịp, lb
V_u	Lực cắt tại tiết diện, lb
V_p	Lực cắt tính theo sức chịu mô men, lb
α	Hệ số cốt thép vượt tải
β_1	Hệ số theo chiều cao phần chịu nén của bê tông
β_d	Trị số tuyệt đối tỷ lệ giữa mô men lớn nhất do tĩnh tải với mô men lớn nhất do toàn bộ tải trọng dọc trục
δ_s	Hệ số mô men lớn nhất khi có lắc lư
δ_{ns}	Hệ số mô men lớn nhất khi không lắc lư
ε_c	Chịu kéo trong bê tông
ε_s	Chịu kéo trong cốt thép
φ	Hệ số giảm tải

III-1. TỔ HỢP TẢI TRỌNG THIẾT KẾ

Tổ hợp tải trọng thiết kế được xác định theo các tải trọng cho trước của kết cấu cần được kiểm tra. Theo Tiêu chuẩnACI 318-99, nếu một kết cấu chỉ chịu tĩnh tải (DL- dead load) và hoạt tải (LL- live load), kiểm tra ứng suất có thể chỉ cần một tổ hợp tải trọng 1.4DL+1.7LL (ACI 9.2.1). Do vậy, việc cộng thêm vào tĩnh tải, hoạt tải chỉ thực hiện khi kết cấu chịu lực gió và động đất, việc xem xét lực gió và động đất sẽ theo cả 2 chiều tác động, theo các tổ hợp tải trọng sau (ACI 9.2).

1,4 DL

1,4 DL + 1,7 LL (ACI 9.2.1)

0,9 DL ± 1,3 WL

0,75 (1,4 DL + 1,7 LL ± 1,7 WL) (ACI 9.2.2)

0,9 DL ± 1,3 * 1,1 EL

0,75 (1,4 DL + 1,7 LL ± 1,7 * 1,1 LL) (ACI 9.2.3)

Ngoài ra cũng có các tổ hợp tải trọng mặc định trong SAP2000 khi dùng tiêu chuẩn ACI 318-99.

Các hệ số giảm hoạt tải có thể được áp dụng cho các thành phần hoạt tải theo từng phân tố để làm giảm thành phần hoạt tải tập trung.

III-2. CÁC HỆ SỐ GIẢM CƯỜNG ĐỘ

Các hệ số giảm cường độ, φ, được áp dụng cho các cường độ danh định nhận được khi thiết kế một cấu kiện. Hệ số φ dùng cho chịu uốn, nén dọc trục, lực cắt và xoắn như sau:

φ = 0,90 cho chịu uốn (ACI 9.3.2.1)

φ = 0,90 cho chịu kéo dọc trục (ACI 9.3.2.2)

φ = 0,90 cho chịu uốn và kéo dọc trục (ACI 9.3.2.2)

φ = 0,75 cho chịu nén dọc trục, chịu uốn và nén dọc trục (cốt thép xoắn trong cột), và (ACI 9.3.2.2)

φ = 0,70 cho chịu nén dọc trục, chịu uốn và nén dọc trục (cốt thép xoắn trong cột), và (ACI 9.3.2.2)

φ = 0,85 cho cắt và xoắn

III-3. THIẾT KẾ CỘT

Không dịch ở đây.

III-4. THIẾT KẾ DẦM

Khi thiết kế dầm bê tông, SAP2000 tính toán và cho biết diện tích cốt thép cần thiết cho chịu uốn và chịu cắt, trên cơ sở mô men uốn, lực cắt trên dầm sinh ra do các tổ hợp tải trọng và các yếu tố khác, được mô tả sau đây. Cốt thép yêu cầu được kiểm tra/thiết kế theo một số mặt cắt xác định dọc dầm.

Tất cả các dầm chỉ được thiết kế theo phương chịu uốn và chịu cắt chính. Tác động của lực dọc trục, hướng uốn phụ, và chịu xoắn có thể có trong dầm phải do người sử dụng tự tính lấy.

Trình tự thiết kế dầm bao gồm các bước sau:

- Thiết kế cốt thép chịu uốn trong dầm
- Thiết kế cốt thép chịu cắt trong dầm

1. Thiết kế cốt thép chịu uốn trong dầm

Cốt thép phía trên và đáy dầm chịu uốn được thiết kế/ kiểm tra theo các mặt cắt dọc nhịp dầm. Khi thiết kế cốt thép chịu uốn cho mô men chính với dầm riêng biệt theo mặt cắt xác định, các bước tính toán như sau:

- Xác định mô men lớn nhất
- Xác định diện tích cột thép

a) Xác định mô men

Khi thiết kế cốt thép chịu uốn của một kết cấu dầm bê tông Thông thường, Trung gian hoặc Đặc biệt, trị số mô men theo mỗi tổ hợp tải trọng ở một mặt cắt xác định tính được theo các trường hợp đặt tải khác nhau với các hệ số tải trọng tương ứng.

Từ đó, mặt cắt dầm được thiết kế theo mô men dương lớn nhất M_u^+ và mô men âm lớn nhất M_u^- tính được theo tất cả các tổ hợp tải trọng.

Mô men âm do cốt thép phía trên chịu. Trường hợp này dầm thường được thiết kế theo mặt cắt hình chữ nhật. Mô men dương do cốt thép phía dưới chịu. Trường hợp này dầm có thể thiết kế theo mặt cắt chữ nhật hoặc chữ T

b) Xác định cốt thép chịu uốn yêu cầu

Trong quá trình thiết kế cốt thép chịu uốn, chương trình tính cả cốt thép chịu kéo và chịu nén. Cốt thép chịu nén được cộng thêm khi mô men thiết kế vượt quá sức chịu mô men lớn nhất của mặt cắt chỉ có một loại cốt thép chịu kéo. Người thiết kế có thể tránh dùng cốt thép chịu nén bằng cách tăng chiều cao có hiệu, chiều rộng của dầm, hoặc tăng mác bê tông.

Trình tự thiết kế dựa trên sơ đồ phân bố ứng suất đơn giản trên hình chữ nhật như đã thể hiện trên hình III-3 (ACI 10,2). Hơn nữa, giả định rằng ứng suất nén trong bê tông chỉ bằng 0,75 lần ứng suất có được trong điều kiện cân bằng kéo-nén (ACI 10,3.3). Khi dùng mô men vượt quá khả năng chịu mô men của dầm trong điều kiện cân bằng này, diện tích cốt thép chịu nén được tính với giả định rằng phần mô men lớn hơn sẽ do cốt thép chịu nén và cộng thêm cốt thép chịu kéo.

Trình tự thiết kế trong SAP2000 cho cả hai trường hợp mặt cắt hình chữ nhật và có cánh (như dầm chữ L và chữ T) được tóm tắt sau đây. Với giả định rằng lực dọc

trục cực đại không vượt quá $0,1f_c'A_g$ (ACI 10,3.3), như vậy tất cả các dầm được thiết kế chỉ theo hướng chịu uốn và chịu cắt chính.

- Thiết kế dầm chữ nhật

Khi thiết kế theo mô men âm và dương, M_u, (tức là tính cốt thép ở trên hoặc dưới dầm) chiều cao phần chịu nén tính theo a (xem hình III-3):

$$a = d - \sqrt{d^2 - \frac{2[M_u]}{0,85f_c'\varphi b}}$$

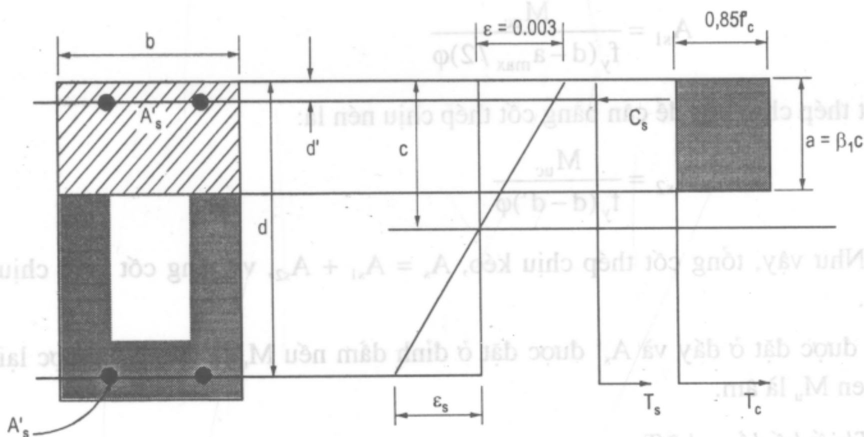
Trong đó, $\varphi = 0,90$ (ACI 9.3.2.1). Các trị số β_1 và c_b tính như sau:

$$\beta_1 = 0,85 - 0,05\left[\frac{f_c' - 4000}{1000}\right], \quad 0,65 \le \beta_1 \le 0,85 \qquad \text{(ACI 10,2.7.3)}$$

$$c_b = \frac{\varepsilon_c E_s}{\varepsilon_c E_s + f_y} d = \frac{87000}{87000 + f_y} d \qquad \text{(ACI 10,2.3, 10,2.4)}$$

Chiều cao cho phép lớn nhất phần chịu nén theo công thức:

$$a_{max} = 0,75\beta_1 c_b \qquad \text{(ACI 10,2.7.1)}$$



Hình III-3: *Thiết kế mặt cắt dầm chữ nhật*

- Nếu $a < a_{max}$, diện tích cốt thép kéo là:

$$A_s = \frac{M_u}{\varphi f_y'(d - a/2)}$$

Cốt thép này đặt ở đáy dầm nếu mô men M_u dương, hoặc ở đỉnh dầm nếu M_u là âm.

- Nếu $a > a_{max}$, cần có cốt thép chịu nén (ACI 10,3.3) và được tính như sau:

+ Lực nén chỉ phát sinh trong bê tông là:

$$C = 0,85\, fc'ba_{max}, \text{ và} \qquad \text{(ACI 10,2.7.1)}$$

Sức chống mô men do bê tông chịu nén và cốt thép chịu kéo là:

$$M_{uc} = C(d - a_{max}/2)\varphi$$

+ Sức chống mô men do cốt thép chịu nén và cốt thép chịu kéo là:

$$M_{us} = M_u - M_{uc}$$

+ Do vậy cốt thép chịu nén yêu cầu được tính là:

$$A_s' = \frac{M_{us}}{f_s'(d - d')\varphi}$$

Trong đó:

$$f_s = 0,003E_s \frac{[c - d']}{c} \qquad \text{(ACI 10,2.4)}$$

+ Cốt thép chịu kéo yêu cầu để cân bằng chịu nén trong bê tônglà:

$$A_{s1} = \frac{M_{uc}}{f_y(d - a_{max}/2)\varphi}$$

và cốt thép chịu kéo để cân bằng cốt thép chịu nén là:

$$A_{s2} = \frac{M_{uc}}{f_y(d - d')\varphi}$$

+ Như vậy, tổng cốt thép chịu kéo, $A_s = A_{s1} + A_{s2}$, và tổng cốt thép chịu nén là A_s'.

A_s được đặt ở đáy và A_s' được đặt ở đỉnh dầm nếu M_u là dương, ngược lại nếu mô men M_u là âm.

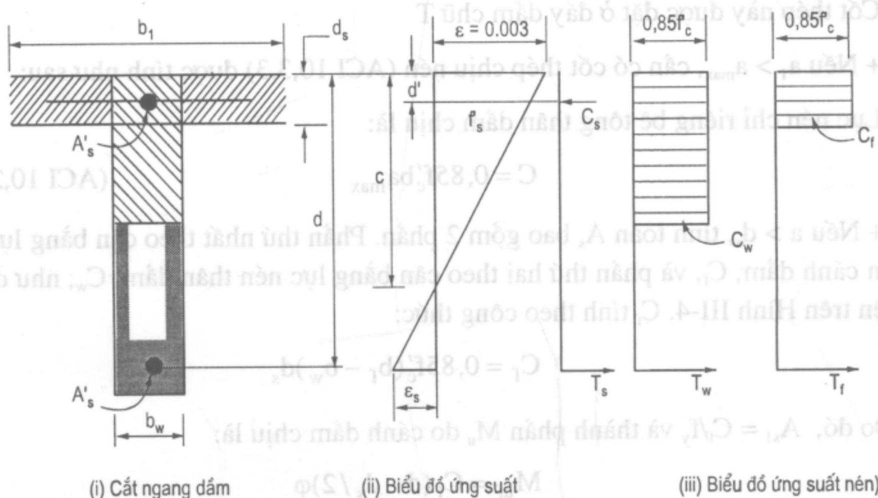
- *Thiết kế dầm chữ T*

Khi thiết kế theo mô men âm, M_u, (tức là cốt thép đặt trên đỉnh dầm), tính toán diện tích cốt thép chính xác tương tự như trên, tức là, số liệu không có dầm T được dùng. Xem hình III-4. Nếu $M_u > 0$, chiều cao phần chịu nén là:

$$a = d - \sqrt{d^2 - \frac{2M_u}{0,85f_c'\varphi b_r}}$$

Chiều cao lớn nhất cho phép vùng chịu nén tính theo công thức:

$$\sigma_{max} = 0,75\,\beta_1 c_b \qquad \text{(ACI 10.2.7.1)}$$



Hình III-4: *Thiết kế dầm chữ T*

- Nếu $a \le d_s$, trình tự tính A_s như với tiết diện chữ nhật đã nêu ở trên. Tuy vậy, trong trường hợp này cánh dầm chịu nén được tính như bề rộng dầm khi tính toán. Hoặc là cốt thép chịu nén tuỳ thuộc trị số $a > a_{max}$.

- Nếu $a > d_s$, tính toán A_s được chia thành 2 phần. Phần thứ nhất theo cân bằng lực nén ở cánh dầm, C_f, và phần thứ hai theo cân bằng lực nén ở thân dầm, C_w, như đã thể hiện trên hình III-4. C_f được tính theo công thức:

$$C_f = 0{,}85f'_c(b_f - b_w)d_s$$

Do đó, $A_{s1} = C_f/f_y$ và thành phần M_u do cánh dầm chịu là:

$$M_{uf} = C_f(d - d/2)\varphi$$

Trong đó $\varphi = 0{,}90$, Do đó, để cân bằng kéo-nén chịu mô men M_u, mô men do cánh dầm chịu được tính theo công thức:

$$M_{uw} = M_u - M_{uf}$$

Thân dầm hình chữ nhật có kích thước b_w và d, chiều cao phần chịu nén là:

$$a_1 = d - \sqrt{d^2 - \frac{2M_{uw}}{0{,}85f'_c\varphi b_w}}$$

+ Nếu $a_1 \le a_{max}$, diện tích cốt thép chịu kéo là:

$$A_{s2} = \frac{M_{uw}}{\varphi f_y(d - a_1/2)}$$

$$A_s = A_{s1} + A_{s2}$$

Cốt thép này được đặt ở đáy dầm chữ T

+ Nếu $a_1 > a_{max}$, cần có cốt thép chịu nén (ACI 10,3.3) được tính như sau:

Lực nén chỉ riêng bê tông thân dầm chịu là:

$$C = 0,85f_c'ba_{max} \qquad \text{(ACI 10,2.7.1)}$$

+ Nếu $a > d_s$, tính toán A_s bao gồm 2 phần. Phần thứ nhất theo cân bằng lực nén phần cánh dầm, C_f, và phần thứ hai theo cân bằng lực nén thân dầm, C_w, như đã thể hhiện trên Hình III-4. C_f tính theo công thức:

$$C_f = 0,85f_c'(b_f - b_w)d_s$$

Do đó, $A_{s1} = C_f/f_y$ và thành phần M_u do cánh dầm chịu là:

$$M_{uf} = C_f(d - d_s/2)\varphi$$

Trong đó $\varphi = 0,90$, Do đó, để cân bằng kéo-nén chịu mô men M_u, mô men do cánh dầm chịu được tính theo công thức:

$$M_{uw} = M_u - M_{uf}$$

Thân dầm hình chữ nhật có kích thước b_w và d, chiều cao phần chịu nén là:

$$a_1 = d - \sqrt{d^2 - \frac{2M_{uw}}{0,85f_c'\varphi b_w}}$$

+ Nếu $a_1 \le a_{max}$, diện tích cốt thép chịu kéo là:

$$A_{s2} = \frac{M_{uw}}{\varphi f_y(d - a_1/2)}$$

$$A_s = A_{s1} + A_{s2}$$

Cốt thép này được đặt ở đáy dầm chữ T

+ Nếu $a_1 > a_{max}$, cần có cốt thép chịu nén (ACI 10,3.3) được tính như sau:

- Lực nén chỉ riêng bê tông thân dầm chịu là:

$$C = 0,85f_c'ba_{max} \qquad \text{(ACI 10,2.7.1)}$$

- Do vậy, sức chịu mô men của bê tông thân dầm và cốt thép chịu kéo là:

$$M_{uc} = c(d - a_{max}/2)\varphi$$

Sức chịu mô men của cốt thép chịu nén và chịu kéo là:

$$M_{us} = M_{uw} - M_{uc}$$

- Do vậy, cốt thép chịu nén được tính là:

$$A_s = \frac{M_{us}}{\varphi f'_s(d - d')}$$

Trong đó: $$f'_s = 0{,}003E_s\frac{[c - d']}{c}$$ (ACI 10.2.4)

- Cốt thép chịu kéo để cân bằng chịu nén với bê tông thân dầm là:

$$A_{s2} = \frac{M_{uc}}{\varphi f_y(d - a_{max}/2)}$$

và cốt thép chịu kéo để cân bằng cốt thép chịu nén là:

$$A_{s3} = \frac{M_{us}}{\varphi f_y(d - d')}$$

Tổng số cốt thép chịu kéo, $A_s = A_{s1} + A_{s2} + A_{s3}$, và tổng số cốt thép chịu nén là A_s'. A_s đặt ở đáy dầm và A_s' đặt ở đỉnh dầm.

- Cốt thép chịu kéo tối thiểu

Cốt thép chịu kéo khi uốn tối thiểu cần có trong dầm chữ nhật khi kết cấu chịu mô men Thông thường, trị số tối thiểu được xét theo 2 giới hạn sau:

$$A_s \geq \max\left[\frac{3\sqrt{f'_c}}{f_y}b_w d \text{ và } \frac{200}{f_y}b_w d\right]$$ (ACI 10.5.1)

hoặc $$A_s \geq \frac{4}{3}A_{s(required)}$$ (ACI 10.5.3)

- Thiết kế đặc biệt có xét đến động đất

Để thiết kế sức chịu mô men đặc biệt của kết cấu bê tông (động đất), dầm được thiết kế thêm các điều kiện sau (xem Bảng III-2):

- Cốt thép dọc tối thiểu sẽ được quy định cho cả phía trên và dưới dầm. Cốt thép cả trên và dưới dầm đều không được nhỏ hơn $A_{s(min)}$ (ACI 21.3.2.1).

$$A_{s(min)} \geq \max\left[\frac{3\sqrt{f'_c}}{f_y}b_w d \text{ và } \frac{200}{f_y}b_w d\right]$$ (ACI 10.5.1)

hoặc $$A_s \geq \frac{4}{3}A_{s(required)}$$ (ACI 10.5.3)

- Cốt thép dầm chịu uốn có giới hạn cực đại là:

$$A_s \geq 0{,}025\, b_w d$$ (ACI 21.3.2.1)

- Tại đầu dầm (gối đỡ), dầm chịu mô men dương (tương ứng với cốt thép đặt ở đáy) sẽ không được nhỏ hơn 1/2 dầm chịu mô men âm (tương ứng với cốt thép đặt ở trên) tại đầu đó (ACI 21.3.2.2)

- Khả năng chịu mô men âm hoặc dương của bất kỳ mặt cắt nào trong dầm không được nhỏ hơn 1/4 khả năng chịu mô men dương hoặc âm ở bất kỳ đầu dầm (gối đỡ) nào (ACI 21.3.2.2).

Với kết cấu bê tông chịu mô men Trung gian (khi thiết kế động đất), dầm phải thoả mãn các điều kiện sau:

- Tại bất kỳ gối đỡ nào, khả năng chịu mô men dương của dầm không được nhỏ hơn 1/3 khả năng chịu mô men âm của dầm tại gối đó (ACI 21.10,4.1).

- Khả năng chịu mô men âm hoặc dương của dầm tại bất kỳ mặt cắt nào trong dầm không được nhỏ hơn 1/5 khả năng chịu mô men dương hoặc âm lớn nhất tại bất kỳ đầu dầm (gối đỡ) nào (ACI 21.10,4.1).

2. Thiết kế cốt thép chịu cắt của dầm

Cốt thép chịu cắt của dầm được thiết kế theo mỗi số hiệu tổ hợp tải trọng do người thiết kế xác định tại các mặt cắt dọc dầm. Khi thiết kế cốt thép chịu cắt cho một đoạn dầm theo một thành phần tổ hợp tải trọng tại một vị trí xác định, theo lực cắt chính của dầm, bao gồm các bước sau:

- Xác định trị số lực cắt, V_u

- Xác định lực cắt, V_c, do bê tông chịu

- Xác định cốt thép yêu cầu trong điều kiện cân bằng kéo-nén

Theo sức chống mô men Trung gian và Đặc biệt (kết cấu chịu uốn), lực cắt thiết kế của dầm cũng dựa trên khả năng chịu mô men danh định và có thể có của câú kiện, tương ứng, cộng thêm hệ số tải trọng thiết kế.

Có 3 mục sau mô tả chi tiết cách tính toán, tương ứng với các chỉ dẫn trên.

a) Xác định lực cắt và mô men

- Khi thiết kế cốt thép chịu cắt của dầm của **một kết cấu bê tông chịu mô men Thông thường**, lực cắt và mô men cho một thành phần tổ hợp tải trọng tại một đoạn dầm được xác định theo trị số lực cắt và mô men liên quan đến tổ hợp tải trọng tương ứng.

- Khi thiết kế **kết cấu bê tông chịu mô men Đặc biệt** (thiết kế động đất), khả năng chịu cắt của dầm được kiểm tra theo lực cắt có thể chịu theo khả năng mô men có thể chịu tại đầu dầm và trọng tải. Kiểm tra này được thực hiện có xét thêm đến yêu cầu kiểm tra thiết kế theo sức chịu mô men Thông thường của dầm. Lực cắt, V_u, được tính theo khả năng có thể chịu mô men của mỗi đầu dầm và lực cắt theo phương trọng lực. Trình tự tính toán lực cắt thiết kế trong dầm theo sức chịu

mô men có thể được cũng giống như đã mô tả trong phần "Thiết kế cột" trang 27 *(nguyên bản, không dịch ở đây).* Chi tiết xem Hình III-2.

Lực cắt thiết kế V_u được tính theo (ACI 21.3.4.1) là:

$$V_u = V_p + V_{D+L} \qquad \text{(ACI 21.3.4.1)}$$

Trong đó, Vp là lực cắt tính được theo khả năng chịu mô men cực hạn tại 2 đầu dầm tác động theo 2 chiều ngược nhau. Do đó, V_p là trị số lớn nhất của V_{p1} và V_{p2}, trong đó

$$V_{p1} = \frac{M_I^- + M_J^+}{L}$$

$$V_{p1} = \frac{M_I^+ + M_J^-}{L}$$

M_I^- = khả năng chịu mô men tại đầu I, với cốt thép chịu kéo ở đỉnh dầm, dùng trị số giới hạn chảy dẻo của thép αf, không có trị số φ (φ =1.0)

M_I^+ = khả năng chịu mô men tại đầu J, với cốt thép chịu kéo ở đáy dầm, dùng trị số giới hạn chảy dẻo của thép αf, không có trị số φ (φ =1.0)

M_J^+ = khả năng chịu mô men tại đầu I, với cốt thép chịu kéo ở đáy dầm, dùng trị số giới hạn chảy dẻo của thép αf, không có trị số φ (φ =1.0)

M_J^- = khả năng chịu mô men tại đầu J, với cốt thép chịu kéo ở đỉnh dầm, dùng trị số giới hạn chảy dẻo của thép αf, không có trị số φ (φ =1.0)

L = Nhịp tĩnh không của dầm.

Với dầm chịu mô men đặc biệt trị số α = 1,25 (ACI R21.3.4.1). V_{D+L} là phần tham gia chịu lực của lực cắt dưới tác dụng của trọng tải phân bố trên nhịp.

- Với **kết cấu chịu mô men trung gian**, khả năng chịu lực cắt của dầm cũng được kiểm tra theo thiết kế lực cắt danh định dựa trên khả năng chịu mô men danh định tại đầu dầm và các yếu tố tải trọng, đồng thời cũng kiểm tra thêm sức chịu mô men Thông thường của dầm. Khi thiết kế lực cắt trong dầm dùng trị số tối thiểu dựa trên sức chịu mô men và lực cắt danh định. Trình tự tính sức chịu mô men danh định (φ = 1,0) tương tự như tính khả năng chịu mô men của kết cấu chịu mô men Đặc biệt, ngoại trừ việc dùng α = 1, ít dùng hơn trị số α = 1,25 (ACI R21.10). Trị số lực cắt dựa trên trị số tải trọng quy định, trừ tải trọng động đất nhân đôi (ACI 21.10,3). Việc tính toán lực cắt trong dầm của một **kết cấu chịu mô men Trung gian** cũng tương tự như tính cột, được mô tả ở trang 28 *(nguyên bản, không dịch ở đây).* Chi tiết xem thêm Bảng III-2.

b) Xác định sức chịu lực cắt của bê tông

Sức chịu lực cắt cho phép của bê tông được tính là:

$$V_c = 2\sqrt{f_c''}b_w d \qquad \text{(ACI 11.3.1.1)}$$

Để thiết kế sức chịu mô men Đặc biệt của kết cấu bê tông, V_c được coi bằng zero nếu cả hai lực nén dọc trục bao gồm cả ảnh hưởng của động đất P_w nhỏ hơn $f_c'A_g/20$ và lực cắt chịu lực động đất V_E nhiều hơn nửa tổng lực cắt lớn nhất có trên toàn bộ chiều dài cấu kiện V_u (tức là $V_E \geq 0{,}5\ V_u$) (ACI 21.3.4.2)

c) Xác định cốt thép chịu cắt yêu cầu

Theo trị số V_u và V_c, cốt thép chịu cắt yêu cầu tính bằng diện tích/đơn vị dài được tính như sau:

$$A_v = \frac{(V_u/\varphi - V_c)s}{f_{ys}d} \qquad \text{(ACI 11.5.6.2)}$$

Sức chống lực cắt do cốt thép được giới hạn như sau:

$$(V_u/\varphi - V_c)s \leq 8\sqrt{f_c'}bd \qquad \text{(ACI 11.5.6.9)}$$

Trong đó, φ, hệ số giảm cường độ là 0,85 (ACI 9.3.2.3). Trị số lớn nhất của A_v, nhận được từ mỗi tổ hợp tải trọng, được cho biết cùng với quá trình kiểm tra lực cắt và liên quan với số tổ hợp tải trọng.

Cốt thép chịu cắt yêu cầu của dầm được thể hiện trong chương trình SAP được dựa trên xem xét cường độ lực cắt. Bất kỳ sốlượng thép đai tối thiểu nào thoả mãn khoảng cách lắp đặt và xem xét khối lượng phải được người thiết kế tính toán độc lập với chương trình.

Bảng III-2. Các thông số thiết kế

Loại kiểm tra/th. kế	Kết cấu chịu mômen thông thường (không động đất)	Kết cấu chịu mô men trung gian (có động đất)	Kết cấu chịu mô men đặc biệt (có động đất)
1	*2*	*3*	*4*
Kiểm tra cột (tương tác)	Số tổ hợp tải trọng quy định	Số tổ hợp tải trọng quy định	Số tổ hợp tải trọng quy định
Thiết kế cột (tương tác)	Số tổ hợp tải trọng quy định $1\% < \rho < 8\%$	Số tổ hợp tải trọng quy định $1\% < \rho < 8\%$	Số tổ hợp tải trọng quy định $\alpha = 1{,}0$, $1\% < \rho < 6\%$
Lực cắt cột	Số tổ hợp tải trọng quy định	Sửa đổi số tổ hợp tải trọng quy định (gấp đôi tải trọng động đất) Khả năng chịu tải của cột. $\varphi = 1{,}0$ và $\alpha = 1{,}0$	Số tổ hợp tải trọng quy định và khả năng chịu cắt của cột $\varphi = 1.0$ và $\alpha = 1.25$

Bảng III-2 (tiếp theo)

1	2	3	4
Thiết kế dầm chịu uốn	Số tổ hợp tải trọng quy định	Số tổ hợp tải trọng quy định	Số tổ hợp tải trọng quy định $\rho \le 0{,}025$ $\rho \ge \frac{3\sqrt{f'_c}}{f_y}$; $\rho \ge \frac{200}{f_y}$
Kiểm tra mômen tối thiểu	Số yêu cầu	$M_{uEND}^{+} \ge \frac{1}{3} M_{uEND}^{-}$ $M_{uSPAN}^{+} \ge \frac{1}{5} \max \times$ $\times [M_u^{+}, M_u^{-}]_{END}$ $M_{uSPAN}^{-} \ge \frac{1}{5} \max \times$ $[M_u^{+}, M_u^{-}]_{END}$	$M_{uEND}^{+} \ge \frac{1}{2} M_{uEND}^{-}$ $M_{uSPAN}^{+} \ge \underline{1} \max \times$ $\times [M_u^{+}, M_u^{-}]_{END}$ $M_{uSPAN}^{-} \ge \underline{1} \max \times$ $\times [M_u^{-}, M_u^{-}]_{END}$
Thiết kế lực cắt dầm	Số tổ hợp tải trọng quy định	Sửa đổi số tổ hợp tải trọng quy định (gấp đôi tải trọng động đất) Khả năng chịu cắt của dầm (Vp) với $\alpha = 1{,}0$ và $\varphi = 1{,}0$ cộng V_{D+L}	Số tổ hợp tải trọng quy định Khả năng chịu cắt của dầm (Vp) với $\alpha = 1{,}25$ và $\varphi = 1{,}0$ cộng V_{D+L} $V_c = 0$

Chương IV

THIẾT KẾ THEO TIÊU CHUẨN AASHTO LRFD 1997

Chương này mô tả chi tiết trình tự thiết kế các kết cấu bê tông khác nhau khi dùng SAP2000 thiết kế theo Tiêu chuẩn của Hội Cầu Đường và Giao thông Mỹ AASHTO LRFD 1997. Các ký hiệu dùng trong chương này xem Bảng IV-1

Thiết kế dựa trên các tổ hợp tải trọng do người dùng quy định. Nhưng chương trình cũng cung cấp các tổ hợp tải trọng mặc định thoả mãn yêu cầu thiết kế hầu hết các dạng kết cấu công trình.

SAP2000 cung cấp các lựa chọn thiết kế hoặc kiểm tra sức chịu mô men của kết cấu thuộc Vùng 1 (động đất thấp), 2, 3 và 4 (động đất cao) theo yêu cầu thiết kế

động đất cho trước. Chi tiết các thông số dùng cho các vùng động đất khác nhau lần lượt được mô tả ở các phần sau.

Hệ đơn vị của Anh SI và MKS được dùng để nhập số liệu. Nhưng chương trình dùng hệ **Inch-Pound-Second**. Để đơn giản, các phương trình trình bày trong chương này đều theo đơn vị **Inch-Pound-Second**, trừ khi có ghi chú khác.

Bảng IV-1. Bảng ký hiệu dùng cho Tiêu chuẩn AASHTO

Ký hiệu	Diễn giải
1	*2*
A_{cv}	Diện tích bê tông dùng để xác định ứng suất cắt, in^2 (sq-in)
A_g	Diện tích toàn bộ bê tông, in^2
A_s	Diện tích cốt thép chịu kéo, in^2
A_s'	Diện tích cốt thép chịu nén, in^2
$A_{s(requied)}$	Diện tích cốt thép chịu kéo yêu cầu, in^2
A_{st}	Tổng diện tích cốt thép dọc cột, in^2
A_v	Diện tích cốt thép chịu cắt, in^2
a	Chiều cao vùng chịu nén, in
a_b	Chiều cao vùng chịu nén trong điều kiện cân bằng, in
a_{max}	Chiều cao lớn nhất cho phép vùng chịu nén, in
b	Bề rộng cấu kiện, in
b_f	Bề rộng chịu tác dụng của cánh dầm (dầm chữ T), in
b_w	Bề rộng thân dầm (dầm chữ T), in
C_m	Hệ số, phụ thuộc vào độ cong của cột, dùng để tính toán hệ số tăng mô men
c	Chiều cao trục trung hoà, in
c_b	Chiều cao trục trung hoà khi điều kiện cân bằng, in
d	Khoảng cách từ mặt cốt thép chịu nén tới chịu kéo, in
d'	Lớp bảo vệ bê tông tính tới tim cốt thép, in
d_s	Bề dày bản bê tông (dầm chữ T), in
E_c	Mô đun đàn hồi của bê tông, psi (p/in^2)
E_s	Mô đun đàn hồi của cốt thép, giả định là 29.000 ksi (= 2.030,000 Kgf/cm^2)
f_c'	Cường độ chịu nén tiêu chuẩn của bê tông, ksi
f_y	Cường độ giới hạn chảy dẻo của cốt thép khi chịu uốn, ksi
f_{ys}	Cường độ giới hạn chảy dẻo của cốt thép khi chịu cắt, ksi
h	Kích thước cột, in

Bảng IV-1 (tiếp theo)

1	2
I_g	Mô men quán tính của toàn tiết diện mặt cắt bê tông quanh trục trung tâm, không tính đến cốt thép, in^4
I_{se}	Mô men quán tính của cốt thép quanh trục trung tâm của cắt ngang cấu kiện, in^4
k	Hệ số ảnh hưởng của chiều dài
L	Chiều dài tĩnh không, in
M_1	Trị số giảm nhỏ mô men cuối cột, kip-in
M_2	Trị số tăng mô men trong cột, kip-in
M_c	Trị số mô men dùng để thiết kế, kip-in
M_b	Trị số mô men cuối của cấu kiện không lắc lư, kip-in
M_s	Trị số mô men cuối của cấu kiện có lắc lư, kip-in
M_u	Trị số mô men tại mặt cắt, kip-in
M_{ux}	Trị số mô men tại mặt cắt quanh trục X, kip-in
M_{uy}	Trị số mô men tại mặt cắt quanh trục Y, kip-in
P_b	Khả năng chịu tải dọc trục trong điều kiện cân bằng chịu kéo, kip
P_e	Cường độ chịu nhiệt Euler của cột, kip
P_{max}	Cường độ chịu tải dọc trục lớn nhất cho phép, kip
P_0	Khả năng chịu tải dọc trục tại điểm lệch tâm zero, kip
P_u	Trị số chịu tải dọc trục tại mặt cắt, kip
r	Bán kính cong của mặt cắt cột, in
V_c	Sức chống cắt của bê tông, kip
V_{D+L}	Sức chống cắt do tải trọng trên nhịp, kip
V_u	Lực cắt tại tiết diện, kip
V_p	Lực cắt tính theo sức chịu mô men, kip
α	Hệ số cốt thép vượt tải
β	Hệ số khả năng gây nứt chéo của bê tông do truyền lực kéo
β_1	Hệ số theo chiều cao phần chịu nén của bê tông
β_d	Trị số tuyệt đối tỷ lệ giữa mô men lớn nhất do tĩnh tải với mô men lớn nhất do toàn bộ tải trọng dọc trục
θ	Góc nghiêng của ứng suất nén xiên với trục dọc của dầm hoặc cột
δ_s	Hệ số mô men lớn nhất khi có lắc lư
δ_{ns}	Hệ số mô men lớn nhất khi không lắc lư
ε_c	Chịu kéo trong bê tông
ε_s	Chịu kéo trong cốt thép
φ	Hệ số giảm tải

IV-1. TỔ HỢP TẢI TRỌNG THIẾT KẾ

Tổ hợp tải trọng thiết kế được xác định theo các tải trọng cho trước của kết cấu cần được kiểm tra. Có nhiều loại tải trọng quy định khác nhau trong quy trình thiết kế hơn là số tổ hợp tải trọng được dùng mặc định. Do vậy người sử dụng phải tự kiểm tra đầy đủ theo các tải trọng và tổ hợp tải trọng quy định.

Có 6 loại tĩnh tải: tĩnh tải của các thành phần cấu kiện và tiếp giáp bởi kết cấu bên ngoài (DC), tĩnh tải do kéo xuống (DD – downdrag), tĩnh tải lớp phủ mặt và phụ kiện (DW), áp lực ngang của đất (ES). Mỗi loại tĩnh tải yêu cầu một hệ số tải trọng riêng biệt.

Có 6 loại hoạt tải: tải trọng xe chạy (LL), tải trọng động cho phép, tải trọng ly tâm xe chạy (CE), tải trọng phanh xe (BR), tải trọng bộ hành (PL), vượt tải do hoạt tải (LS). Tất cả các tải trọng đó cùng một hệ số nên không cần tính riêng biệt.

Nếu kết cấu chịu tĩnh tải (DL), hoạt tải (LL), lực gió (WL) và động đất (EL), việc xem xét lực gió và động đất sẽ theo cả 2 chiều, các tổ hợp tải trọng mặc định được xem xét như sau theo Cường độ và trạng thái Cực hạn (AASHTO 3.4.1).

1,5 DL	(Cường độ- IV)
1.25 DL + 1,25 LL	(Cường độ- I)
0,9 DL ± 1,4 WL	(Cường độ- III)
1.25 DL ± 1,4 WL)	(Cường độ- III)
1.25 DL + 1,35 LL ± 0,40 WL	(Cường độ- V)
0,9 DL ± 1,0 EL	(Cực hạn-I)
1,25 DL + 0,5 LL ± 1,0 EL)	(Cực hạn-I)

Ngoài ra cũng có các tổ hợp tải trọng mặc định trong SAP2000 khi dùng tiêu chuẩn AASHTO LRFD 1997. Người sử dụng xác định thêm các tổ hợp tải trọng cần thiết khác.

Các hệ số giảm hoạt tải có thể được áp dụng cho các thành phần hoạt tải theo từng phân tố để làm giảm thành phần hoạt tải tập trung.

IV-2. CÁC HỆ SỐ GIẢM CƯỜNG ĐỘ

Các hệ số giảm cường độ, φ, được áp dụng cho các cường độ danh định nhận được khi thiết kế một cấu kiện. Hệ số φ dùng cho chịu uốn, nén dọc trục, lực cắt và xoắn như sau:

φ = 0,90 cho chịu uốn	(AASHTO 5.5.4..2.1)
φ = 0,90 cho chịu kéo dọc trục	(AASHTO 5.5.4..2.1)

$\varphi = 0{,}90$ cho chịu uốn và kéo dọc trục (AASHTO 5.5.4..2.1)

$\varphi = 0{,}90$ cho chịu cắt và xoắn (AASHTO 5.5.4..2.1)

$\varphi = 0{,}75$ chỉ chịu nén dọc trục, hoặc chịu nén dọc trục và chịu uốn (AASHTO 5.5.4..2.1)

$\varphi = 0{,}50$ chỉ chịu nén dọc trục, hoặc chịu nén dọc trục và uốn trong vùng động đất 3 và 4 (AASHTO 5.5.4..2.3, AASHTO 5.10,11.4.1b)

Trị số φ bao gồm nén dọc trục và uốn thay đổi từ 0,75 đến 0,90 dựa trên tải trọng dọc trục. Theo trị số thấp của tải trọng dọc trục, trị số φ tăng tuyến tính từ 0,75 tới 0,90 trong khi tải trọng dọc trục giảm từ 0,1 $f_c'A_g$ tới zero (AASHTO 5.5.4.2.1). Để thiết kế trong vùng động đất 3 và 4, trị số φ bao gồm nén dọc trục và uốn thay đổi từ 0,5 tới 0,9 tuỳ theo tải trọng dọc trục. Theo trị số thấp của tải trọng dọc trục, trị số φ tăng tuyến tính từ 0,5 tới 0,9 trong khi tải trọng dọc trục giảm từ 0,2 $f_c'A_g$ tới zero (AASHTO 5.10,11.4.1b). Trường hợp bao gồm kéo dọc trục, φ thường bằng 0,9 (AASHTO 5.5.4.2.1).

IV-3. THIẾT KẾ CỘT

Không dịch ở đây.

IV-4. THIẾT KẾ DẦM

Khi thiết kế dầm bê tông, SAP2000 tính toán và cho biết diện tích cốt thép cần thiết cho chịu uốn và chịu cắt, trên cơ sở mô men uốn, lực cắt trên dầm sinh ra do các tổ hợp tải trọng và các yếu tố khác, được mô tả sau đây. Cốt thép yêu cầu được kiểm tra/ thiết kế theo một số mặt cắt xác định dọc dầm.

Tất cả các dầm chỉ được thiết kế theo phương chịu uốn và chịu cắt chính. Tác động của lực dọc trục, hướng uốn phụ, và chịu xoắn có thể có trong dầm phải do người sử dụng tự tính lấy.

Trình tự thiết kế dầm bao gồm các bước sau:

- Thiết kế cốt thép chịu uốn trong dầm

- Thiết kế cốt thép chịu cắt trong dầm

1. Thiết kế cốt thép chịu uốn trong dầm

Cốt thép phía trên và đáy dầm chịu uốn được thiết kế/ kiểm tra theo các mặt cắt dọc nhịp dầm. Khi thiết kế cốt thép chịu uốn cho mô men chính với dầm riêng biệt theo mặt cắt xác định, các bước tính toán như sau:

- Xác định mô men lớn nhất

- Xác định diện tích cốt thép

a) Xác định mô men

Khi thiết kế cốt thép chịu uốn của một kết cấu dầm bê tông, trị số mô men theo mỗi tổ hợp tải trọng ở một mặt cắt xác định tính được theo các trường hợp đặt tải khác nhau với các hệ số tải trọng tương ứng. Mặt cắt dầm được thiết kế theo mô men dương lớn nhất M_u^+ và mô men âm lớn nhất M_u^- tính được theo tất cả các tổ hợp tải trọng.

Mô men âm do cốt thép phía trên chịu. Trường hợp này dầm thường được thiết kế theo mặt cắt hình chữ nhật. Mô men dương do cốt thép phía dưới chịu. Trường hợp này dầm có thể thiết kế theo mặt cắt chữ nhật hoặc chữ T

b) Xác định cốt thép chịu uốn yêu cầu

Trong quá trình thiết kế cốt thép chịu uốn, chương trình tính cả cốt thép chịu kéo và chịu nén. Cốt thép chịu nén được cộng thêm khi mô men thiết kế vượt quá sức chịu mô men lớn nhất của mặt cắt chỉ có một loại cốt thép chịu kéo. Người thiết kế có thể tránh dùng cốt thép chịu nén bằng cách tăng chiều cao có hiệu, chiều rộng của dầm, hoặc tăng mác bê tông.

Trình tự thiết kế dựa trên sơ đồ phân bố ứng suất đơn giản trên hình chữ nhật như đã thể hiện trên hình IV-3 (AASHTO 5.7). Hơn nữa, giả định rằng chiều cao lớn nhất phần chịu nén bằng 0,42d (AASHTO 5.7.3.3.1). Khi dùng mô men vượt quá khả năng chịu mô men của dầm trong điều kiện giới hạn này, diện tích cốt thép chịu nén được tính với giả định rằng phần mô men lớn hơn sẽ do cốt thép chịu nén và cộng thêm cốt thép chịu kéo.

Khi thiết kế cốt thép dầm chịu uốn, các giới hạn sau được dùng với cường độ cốt thép chịu kéo và cường độ bê tông chịu nén:

$$f_c' \le 10 \text{ ksi} \qquad \text{(AASHTO 5.1, 5.4.2.1)}$$

$$f_y \le 75 \text{ ksi} \qquad \text{(AASHTO 5.4.3.1)}$$

Trình tự thiết kế trong SAP2000 cho cả hai trường hợp mặt cắt hình chữ nhật và có cánh (như dầm chữ L và chữ T) được tóm tắt sau đây. Tất cả các dầm được thiết kế chỉ theo hướng chịu uốn và chịu cắt chính.

- Thiết kế dầm chữ nhật

Khi thiết kế theo mô men âm hoặc dương, M_u, (tức là tính cốt thép ở trên hoặc dưới dầm) chiều cao phần chịu nén tính theo a (xem hình IV-3):

$$a = d - \sqrt{d^2 - \frac{[M_u]}{0{,}85 f_c' \varphi b}} \qquad \text{(AASHTO 5.7.2.1)}$$

Trong đó, φ = 0,90 (AASHTO 5.5.4.2.1) như trên. Các trị số β_1 và c_b tính như sau:

$$\beta_1 = 0,85 - 0,05\,[f_c' - 4], \quad 0,65 \le \beta_1 \le 0,85 \qquad \text{(AASHTO 5.7.2.2)}$$

$$C_b = \frac{\varepsilon_c E_s}{\varepsilon_c E_s + f_y} d = \frac{87}{87 + f_y} d \qquad \text{(AASHTO 5.7.2.1)}$$

Chiều cao cho phép lớn nhất phần chịu nén theo công thức:

$$a_{max} = \min\,[\,\beta_1 c_b\,,\ 0,42\,d\,] \qquad \text{(AASHTO 5.7.2.2, 5.7.3.3.1)}$$

- Nếu a ≤ a_{max} (AASHTO 5.7.3.3.1), diện tích cốt thép kéo là:

$$A_s = \frac{M_u}{\varphi f_y (d - a/2)}$$

Cốt thép này đặt ở đáy dầm nếu mô men M_u dương, hoặc ở đỉnh dầm nếu M_u là âm.

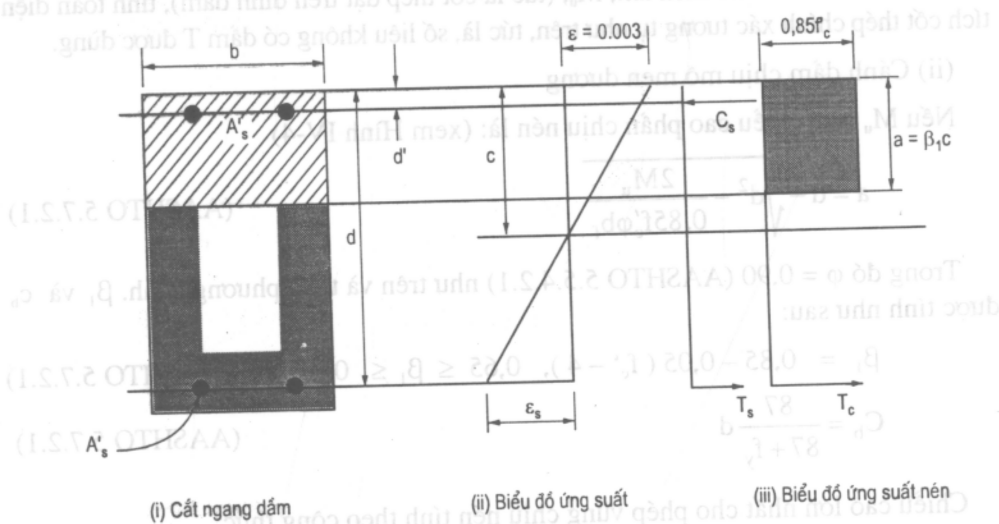
- Nếu a > a_{max} (AASHTO 5.7.3.3.1), cần có cốt thép chịu nén (ACI 10,3.3) và được tính như sau:

+ Lực nén chỉ phát sinh trong bê tông là:

$$C = 0,85\,f_c' b a_{max}, \text{ và} \qquad \text{(AASHTO 5.7.2.2)}$$

Sức chống mô men do bê tông chịu nén và cốt thép chịu kéo ở đáy là:

$$M_{uc} = C\,(d - a_{max}/2\,)\varphi$$



Hình IV-3: Thiết kế mặt cắt dầm chữ nhật

+ Sức chống mô men do cốt thép chịu nén và cốt thép chịu kéo là:

$$M_{us} = M_u - M_{uc}$$

+ Do vậy cốt thép chịu nén yêu cầu được tính là:

$$A'_s = \frac{M_{us}}{(f'_s - f'_c)(d - d')\varphi}$$

Trong đó: $$f'_s = 0,003E_s \frac{[c - d']}{c}$$ (AASHTO 5.7.2.1)

+ Cốt thép chịu kéo yêu cầu để cân bằng chịu nén trong bê tônglà:

$$A_{s1} = \frac{M_{uc}}{f_y(d - a_{max}/2)\varphi}$$

và cốt thép chịu kéo để cân bằng cốt thép chịu nén là:

$$A_{s2} = \frac{M_{us}}{f_y(d - d')\varphi}$$

+ Như vậy, tổng cốt thép chịu kéo, $A_s = A_{s1} + A_{s2}$, và tổng cốt thép chịu nén là A_s'.

A_s được đặt ở đáy và A_s' được đặt ở đỉnh dầm nếu M_u là dương, ngược lại nếu mômen M_u là âm.

- *Thiết kế dầm chữ T*

(i) Cánh dầm chịu mô men âm

Khi thiết kế theo mô men âm, M_u, (tức là cốt thép đặt trên đỉnh dầm), tính toán diện tích cốt thép chính xác tương tự như trên, tức là, số liệu không có dầm T được dùng.

(ii) Cánh dầm chịu mô men dương

Nếu $M_u > 0$, chiều cao phần chịu nén là: (xem Hình IV-4)

$$a = d - \sqrt{d^2 - \frac{2M_u}{0,85f'_c\varphi b_f}}$$ (AASHTO 5.7.2.1)

Trong đó $\varphi = 0,90$ (AASHTO 5.5.4.2.1) như trên và theo phương trình. β_1 và c_b được tính như sau:

$$\beta_1 = 0,85 - 0,05(f_c' - 4), \quad 0,65 \le \beta_1 \le 0,85$$ (AASHTO 5.7.2.1)

$$C_b = \frac{87}{87 + f_y}d$$ (AASHTO 5.7.2.1)

Chiều cao lớn nhất cho phép vùng chịu nén tính theo công thức:

$$a_{max} = \min[\beta_1 c_b, 0,42d]$$ (AASHTO 5.7.2.2, 5.7.3.3.1)

- Nếu $a \le \beta_1 d_s$ (AASHTO 5.7.3.2.2), trình tự tính A_s như với tiết diện chữ nhật đã nêu ở trên. Tuy vậy, trong trường hợp này b_f được coi như bề rộng dầm. Hoặc là cốt thép chịu nén tuỳ thuộc trị số $a > a_{max}$.

- Nếu $a > \beta_1 d_s$ (AASHTO 5.7.3.2.2), tính toán A_s được chia thành 2 phần. Phần thứ nhất theo cân bằng lực nén ở cánh dầm, C_f, và phần thứ hai theo cân bằng lực nén ở thân dầm, C_w, như đã thể hiện trên hình IV-4.

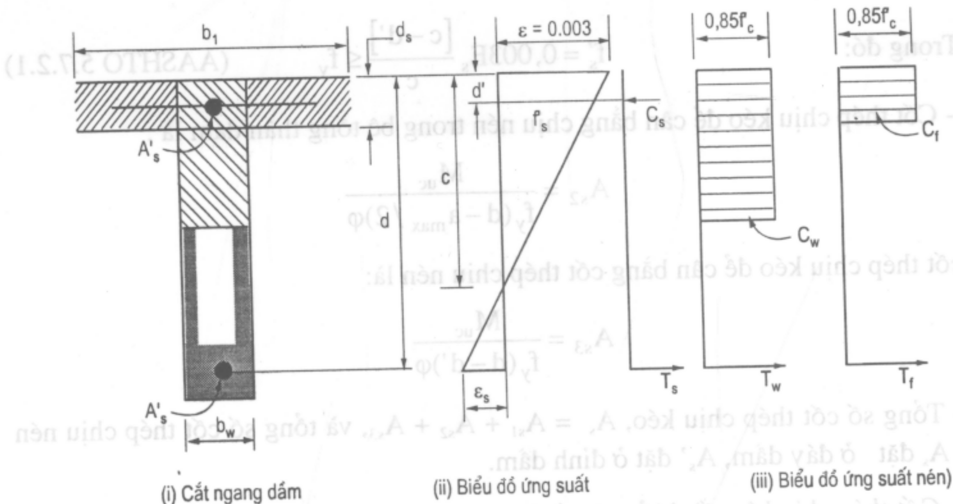
$$C_f = 0,85f_c'(b_f - b_w)\beta_1 d_s \qquad \text{(AASHTO 5.7.3.2.2)}$$

Do đó, $A_{s1} = C_f/f_y$ và thành phần M_u do cánh dầm chịu là:

$$M_{uf} = C_f(d - \beta_1 d_s/2)\varphi$$

Như vậy, để cân bằng kéo-nén chịu mô men M_u, mô men do thân dầm chịu được tính theo công thức:

$$M_{uw} = M_u - M_{uf}$$



Hình IV-4: *Thiết kế dầm chữ T*

Thân dầm hình chữ nhật có kích thước b_w và d, chiều cao thiết kế phần chịu nén là hình chữ nhật:

$$a = d - \sqrt{d^2 - \frac{2M_u}{0,85f_c'\varphi b_f}}$$

+ Nếu $a_1 \le a_{max}$ (AASHTO 5.7.3.3.1), diện tích cốt thép chịu kéo là:

$$A_{s2} = \frac{M_{iw}}{\varphi f_y (d - a_1/2)}$$

$$A_s = A_{s1} + A_{s2}$$

Cốt thép này được đặt ở đáy dầm chữ T

+ Nếu $a_1 > a_{max}$ (AASHTO 5.7.3.3.1), cần có cốt thép chịu nén được tính như sau:

- Lực nén chỉ riêng bê tông thân dầm chịu là:

$$C = 0{,}85 f_c' b a_{max} \qquad \text{(AASHTO 5.7.2.2)}$$

- Do đó, sức chịu mô men do thân dầm bê tông và cốt thép chịu kéo là:

$$M_{uc} = C\,(d - a_{max}/2)\varphi$$

sức chịu mô men do cốt thép chịu nén và chịu kéo là:

$$M_{us} = M_{uw} - M_{uc}$$

- Do đó, cốt thép chịu nén được tính theo công thức:

$$A_s' = \frac{M_{us}}{(f_s' - \alpha f_c')(d - d')\varphi}$$

Trong đó: $$f_s' = 0{,}003 E_s \frac{[c - d']}{c} \le f_y \qquad \text{(AASHTO 5.7.2.1)}$$

- Cốt thép chịu kéo để cân bằng chịu nén trong bê tông thân dầm là :

$$A_{s2} = \frac{M_{uc}}{f_y (d - a_{max}/2)\varphi}$$

và cốt thép chịu kéo để cân bằng cốt thép chịu nén là:

$$A_{s3} = \frac{M_{uc}}{f_y (d - d')\varphi}$$

- Tổng số cốt thép chịu kéo, $A_s = A_{s1} + A_{s2} + A_{s3}$, và tổng số cốt thép chịu nén A_s'. A_s đặt ở đáy dầm, A_s' đặt ở đỉnh dầm.

- *Cốt thép chịu kéo tối thiểu và tối đa*

Cốt thép chịu kéo khi uốn theo nhiệt độ tối thiểu và bê tông co ngót trong tiết diện chữ nhật là:

$$A_s \ge \min\left[\frac{0{,}11}{f_y} b_w d \ \text{ và } \ 0{,}0015 b_w d\right] \qquad \text{(AASHTO 5.10,8.2)}$$

hoặc cốt thép chịu kéo khi uốn tối thiểu để tránh bị hư hỏng sớm do chịu uốn trong tiết diện chữ nhật là:

$$A_s \geq \min\left[\frac{0,03f'_c}{f_y}b_w d \ \text{ và } \ \frac{4}{3}A_{s(required)}\right] \quad \text{(AASHTO 5.7.3.3.2)}$$

hoặc giới hạn trên của 0,04 lần diện tích toàn bộ thân dầm với toàn bộ cốt thép chịu kéo và chịu nén, yêu cầu như sau:

$A_s \leq 0{,}04\ bd$ dầm chữ nhật

$\leq 0{,}04\ b_w d$ dầm chữ T

$A_s' \leq 0{,}04\ bd$ dầm chữ nhật

$\leq 0{,}04\ b_w d$ dầm chữ T

2. Thiết kế cốt thép chịu cắt của dầm

Cốt thép chịu cắt của dầm được thiết kế theo mỗi số hiệu tổ hợp tải trọng do người thiết kế xác định tại các mặt cắt dọc dầm. Khi thiết kế cốt thép chịu cắt cho một đoạn dầm theo một thành phần tổ hợp tải trọng tại một vị trí xác định, theo lực cắt chính của dầm, bao gồm các bước sau:

- Xác định trị số các lực tác dụng lên mặt cắt, M_u và V_u . Lưu ý rằng M_u cần thiết để tính v_c .

- Xác định ứng suất cắt, v_c, do riêng bê tông chịu

- Xác định cốt thép yêu cầu trong điều kiện cân bằng kéo-nén

Theo sức chống mô men trong vùng động đất 3 và 4, lực cắt thiết kế của dầm cũng dựa trên khả năng vượt quá sức chịu mô men có thể có của cấu kiện, cộng thêm hệ số mô men (AASHTO 3.10,9.4.3).

Sau đây mô tả chi tiết cách tính toán, tương ứng với các chỉ dẫn trên.

a) Xác định lực cắt và mô men

- Khi thiết kế cốt thép chịu cắt của dầm của **một kết cấu bê tông chịu mô men trong Vùng động đất 1 và 2**, lực cắt và mô men cho một thành phần tổ hợp tải trọng tại một đoạn dầm được xác định theo trị số lực cắt và mô men liên quan đến tổ hợp tải trọng tương ứng.

- Khi thiết kế lực cắt của **kết cấu bê tông chịu mô men trong Vùng động đất 3 và 4 (thiết kế động đất)**, tiếp theo cần kiểm tra thêm yêu cầu khả năng chịu mô men của kết cấu trong Vùng 1 và 2. Khi thiết kế sức chịu mô men của kết cấu bê tông trong vùng 3 và 4, lực cắt thiết kế trong một dầm, V_u, cũng tính toán từ khả năng vượt quá sức chịu mô men (AASHTO 3.10,9.4.3). Thiết kế lực cắt V_u như sau:

$$V_u = V_p + V_{D+L} \quad \text{(AASHTO 3.10,9.4.3)}$$

Trong đó V_p là lực cắt có được theo tính toán vượt cường độ mô men cực hạn tại 2 đầu dầm tác dụng theo chiều ngược nhau. Do đó, V_p là lớn nhất của V_{p1} và V_{p2}, trong đó

$$V_{p1} = \frac{M_I^- + M_J^+}{L}$$

$$V_{p2} = \frac{M_I^+ + M_J^-}{L}$$

Trong đó

M_I^+, M_I^- = khả năng chịu mô men dương và âm tại đầu I, có được bởi nhân sức chịu danh định với 1.3;

M_J^+, M_J^- = khả năng chịu mô men dương và âm tại đầu J, có được bởi nhân sức chịu danh định với 1.3, và

L - Nhịp tĩnh không của dầm.

V_{D+L} là sức chịu lực cắt từ các tải trọng phân bố trong dầm. Chi tiết xem bảng IV-2.

b) Xác định sức chịu lực cắt của bê tông

Có trị số lực thiết kế M_u và V_u, sức chịu lực cắt của riêng bê tông, v_c, được tính như sau:

$$v_c = 0,0316\beta\sqrt{f_c'} \qquad \text{(AASHTO 5.8.3.3)}$$

Trong đó, β là hệ số cho biết khả năng nứt chéo của bê tông do chuyền lực kéo. Đó là một hàm số của điều kiện ứng suất và trị số gần đúng là 2.0 (AASHTO 5.8.3.4.1). Trị số đó được tính theo AASHTO Bảng 5.8.3.4.2-1 qua một trình tự lặp lại.

c) Xác định cốt thép chịu cắt yêu cầu

- Ứng suất cắt trung bình tính cho tiết diện chữ nhật là:

$$v = \frac{V_u}{b_w d}$$

Theo các loại mặt cắt khác $b_w d$ được thay bằng A_{cv}, diện tích chịu cắt thể hiện trên hình IV-2.

- Ứng suất cắt trung bình, v, bị giới hạn bởi trị số lớn nhất, v_{max}, được tính là:

$$v_{max} = 0,25f_c' \qquad \text{(AASHTO 5.8.3.3)}$$

- Cốt thép chịu cắt trên một đơn vị khoảng cách được tính như sau:

Nếu $v \le \varphi\,(v_c/2)$,

$$\frac{A_v}{s} = 0 \qquad \text{(AASHTO 5.8.2.4)}$$

hoặc là nếu $\varphi(v_c/2) < v \le \varphi\,(v_c + 0,0316\sqrt{f_c'}\cot\theta)$.

$$\frac{A_v}{s} = \frac{0,0316\sqrt{f_c'}\,b_w}{f_{yh}}$$ (AASHTO 5.8.2.5)

hoặc là nếu $\varphi\,(v_c + 0,0316\sqrt{f_c'}\cot\theta) < v \le \varphi v_{max}$

$$\frac{A_v}{s} = \frac{(v - \varphi v_c)b_w}{\varphi f_{yh}\cot\theta}$$ (AASHTO 5.8.3.3)

hoặc là nếu $v > v_{max}$ xác định là không đạt yêu cầu (AASHTO 5.8.3.3)

Ở đây θ là góc nghiêng của ứng suất nén theo phương chéo. Đó là một hàm số của điều kiện ứng suất và có giá trị gần đúng là 45 độ (AASHTO 5.8.3.4.1). Được tính theo AASHTO Bảng 5.8.3.4.2-1) bằng phương pháp lặp lại. Ở đây mặc định trị số giảm cường độ φ = 0,90 (AASHTO 5.5.4.2.1)

Trị số giới hạn f_{yh} là 60 ksi cho mọi kết cấu

$$f_{yh} \le 60\text{ ksi}$$ (AASHTO 5.8.2.8)

Trị số giới hạn f_c' là 10 ksi cho mọi vùng động đất

$$f_c' \le 10\text{ ksi}$$ (AASHTO 5.1, 5.4.2.1)

Tất cả trị số lớn nhất A_v/s, nhận được từ mỗi tổ hợp tải trọng, được đưa ra theo hướng chính và phụ dọc cột với việc kiểm tra lực cắt và liên quan đến số tổ hợp tải trọng.

Yêu cầu cốt thép chịu cắt của dầm do chương trình đưa ra chỉ thuần tuý dựa trên các điều kiện nêu trên. Bất kỳ yêu cầu cốt đai tối thiểu nào khác để thoả mãn khoảng cách/ hoặc thể tích lắp đặt phải do người sử dụng tự xác định.

Bảng IV-2. Các thông số thiết kế

Loại kiểm tra/th. kế	Kết cấu chịu mômen Vùng 1 và 2 (không động đất)	Kết cấu chịu mô men vùng 3 và 4 (có động đất)
1	*2*	*3*
φ khi chịu nén	$\varphi = 0,75$ nếu $P_u > 0,10\,f_c'$	$\varphi = 0,50$, nếu $P_u > 0,20\,f_c'$
Kiểm tra cột (tương tác)	Số tổ hợp tải trọng quy định	Số tổ hợp tải trọng quy định
Thiết kế cột (tương tác)	Số tổ hợp tải trọng quy định $0,135\,f_c'/fy < \rho < 0,08$	Số tổ hợp tải trọng quy định $0,01 < \rho < 0,06$
Lực cắt cột	Số tổ hợp tải trọng quy định $v_c = 0,0316\beta\sqrt{f_c'}$	Số tổ hợp tải trọng quy định Khả năng chịu cắt của cột (V_p) với hệ số vượt tải 1.3 $v_c = 0,0316\beta\sqrt{f_c'}$ nếu $P_u \ge 0,1\,fc'A_g$

Bảng IV-2 (tiếp theo)

1	*2*	*3*
Lực cắt cột	Thể tích cốt thép chịu cắt tối thiểu trong khớp dẻo chỉ cho Vùng 2	$v_c = 0,0316\beta\sqrt{f'_c}\,\frac{P_u}{0,1f'_cA_g} \geq 0$ trừ khi có quy định khác. Thể tích cốt thép chịu cắt tối thiểu trong khớp dẻo chỉ cho Vùng 3 và 4
Thiết kế dầm (chịu uốn)	Số tổ hợp tải trọng quy định	Số tổ hợp tải trọng quy định $\rho \leq 0,025$ $\rho \geq \frac{3\sqrt{f'_c}}{f_y}$ $\rho \geq \frac{200}{f_y}$
Cốt thép chịu uốn tối thiểu của dầm	$\frac{0,11}{f_y}b_wd,\ 0,0015b_wd$ $0,03\frac{f'_c}{f_f},\ \frac{4}{3}S_{s(require)}$	$\frac{0,11}{f_y}b_wd,\ 0,0015b_wd$ $0,03\frac{f'_c}{f_f},\ \frac{4}{3}S_{s(require)}$
Thiết kế dầm (chịu cắt)	Số tổ hợp tải trọng quy định $v_c = 0,0316\beta\sqrt{f'_c}$	Số tổ hợp tải trọng quy định Khả năng chịu cắt của dầm (V_p) với hệ số vượt tải 1.3 $v_c = 0,0316\beta\sqrt{f'_c}$

TÀI LIỆU THAM KHẢO

1. KS Hoàng Hồng. *Lập trình Turbo Pascal 7.0.*
   Nhà xuất bản Giao thông vận tải – 1999
2. Nguyễn Đình Tê, Hoàng Đức Hải. *Giáo trình Lý thuyết và Bài tập Pascal.*
   Nhà xuất bản Giáo dục 1999.
3. Trần Quang Vinh. *Cấu trúc máy vi tính.*
   Nhà xuất bản Giáo dục 1999.
4. Đặng Hữu, Đỗ Bá Chương, Nguyễn Xuân Trục. *Sổ tay Thiết kế đường ôtô.*
   Nhà xuất bản Khoa học Kỹ thuật 1976.
5. Lê Quý An, Nguyễn Công Mẫn, Nguyễn Văn Quỳ. *Cơ học đất.*
   Nhà xuất bản Đại học và Trung học chuyên nghiệp 1970.
6. KS Doãn Hoa. *Thiết kế đường ô tô - Đường ngoài đô thị và đường đô thị* – Tập 1, Tập 2.
   Nhà xuất bản Xây dựng 1999-2000.

MỤC LỤC

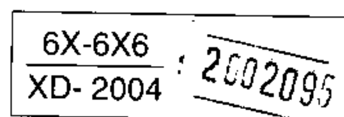
MỘT SỐ CHƯƠNG TRÌNH PASCAL ĐƠN GIẢN CHO THIẾT KẾ ĐƯỜNG ÔTÔ

Chịu trách nhiệm xuất bản:

BÙI HỮU HẠNH

Biên tập:	LƯƠNG XUÂN HỘI
Chế bản:	TRẦN KIM ANH
Trình bày bìa:	NGUYỄN HỮU TÙNG
Sửa bản in:	LƯƠNG XUÂN HỘI
	DOÃN HOA

In 500 cuốn, khổ 17 × 24cm, tại Xưởng in Nhà xuất bản Xây dựng. Giấy chấp nhận đăng kí kế hoạch xuất bản số 326/XB-QLXB-5 ngày 18-3-2004. In xong và nộp lưu chiểu tháng 5-2004.



Giá : 40.000đ